INTRODUCERE ÎN REȚELELE DE CALCULATOARE

GHIDUL TĂU PRACTIC ÎN CISCO NETWORKING

Ramon Nastase

#InvataRetelistica - Introducere in Retele de Calculatoare

In primul rand vreau sa te felicit si sa iti multumesc pentru faptul ca ai luat **decizia de a investii in tine** si de a devenii mai bun.

De la acest ghid te poti astepta sa intelegi:

- cum functioneaza **Internetul**,
- cum **comunica** Dispozitivele (telefon, laptop, server etc) in Internet
- cum functioneaza **retelele** si de cate **tipuri** sunt,
- ce este un **Router, Switch**, adresa **IP**, adresa **MAC**, etc.
- si cum le poti **configura** pentru a obtine conectivitate end-to-end.
- moduri de rutare (**Statica si Dinamica**)
- concepte de Switching (**VLAN-uri, Trunk-uri, RoaS**)
- filtrarea traficului prin **ACL-uri**
- folosirea tehnologiei **NAT** pentru accesarea Internetului

Aceasta carte este structurata in 16 capitole mari care cuprind diferite teme, apartinand bazelor retelisticii. Aceasta carte acopera o parte din materia **modulelor 1 si 2** ale cursului CCNA. Daca iti doresti o cariera in Retele de Calculatoare, atunci iti recomand sa te axezi pe obtinerea acestei certificari, iar cea mai buna varianta este sa-ti dai certificarea **CCENT** (din materia CCNA 1 & 2).

Iata cateva resurse pentru tine legate de certificare:

- Informatii despre **CCENT** si **CCNA**
- Centru de certificare autorizat Cisco - **Pearson VUE**

Iti urez spor la treaba, iar daca ai intrebari nu ezita sa ma contactezi pe **email**, **Facebook** sau **YouTube**.

Cuprins

Capitolul 1 - Elemente de Baza despre Retele

O **retea** reprezinta un **ansamblu de dispozitive** (PC-uri, Routere, Switch-uri etc.) interconectate, care pot comunica (schimba informatii) intre ele.

1) Tipuri de retele

Retelele pot fi de mai multe tipuri:

- **LAN –** *Local Area Network* – ex: reteaua ta (prin cablu) de acasa
- **MAN –** *Metropolitan Area Network* – ex: reteaua extinsa, pe suprafata unui oras
- **WAN –** *Wide Area Network* – ex: Internetul
- **WLAN –** *Wireless LAN* – reteaua ta, wireless, de acasa

Pe langa aceste tipuri de retele de calculatoare mai exista si altele de diferite dimensiuni sau care au total alte scopuri (ex: SAN - Storage Area Network).

O retea de tip **LAN** este o retea relativ mica care este locala unei organizatii (si de obicei limitata din punct de vedere geometric). Spre exemplu reteaua ta de acasa este considerata o retea LAN pentru ca este limitata din punctul de vedere al numarului de dispozitive conectate la ea. O retea a unei scoli (desi este mai mare) va fi considerata tot LAN pentru ca leaga toate calculatoarele, serverele etc. in aceeasi retea si sunt limitati din punct de vedere geografic.

Daca ar fi sa combinam mai multe retele si sa permitem interconectarea lor pe raza unui oras, atunci am forma un **MAN** (o retea mai extinsa, la nivelul unui oras, care ofera viteza mai ridicate de transfer de date fata de viteza “la net obijnuita”).

O retea **WAN** este o retea este o retea de retele pentru ca se afla pe o zona geografica mai mare fata de cele enumerate mai devreme si permite extinderea pe mai multe orase, tari sau chiar continente.

Asadar, dupa cum am spus si mai devreme, o retea MAN interconecteaza mai multe retele din raza unui oras (metropole chiar, cum ar fi Bucurestiul), iar scopul unei retele de tip WAN este acela de a interconecta retelele mai multor orase (sau chiar al mai multor tari). Astfel, se va forma o retea foarte mare, extinsa pe o suprafata geografica mare. De asemenea, putem spune ca mai multe retele LAN interconectate formeaza o retea WAN.

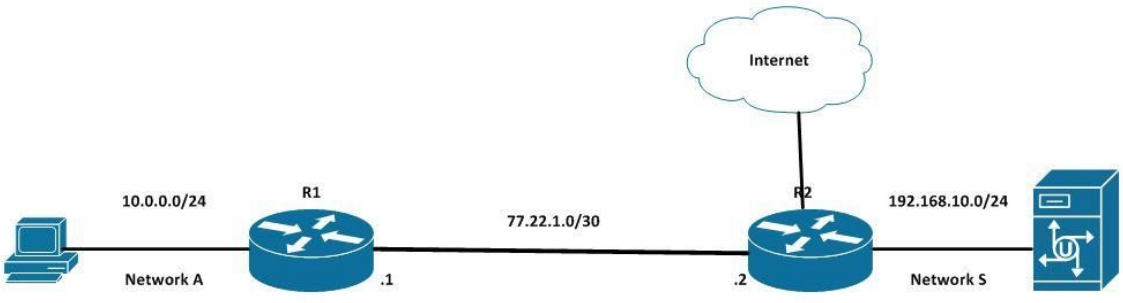


Figura 1.1

De exemplu, in figura 1.1 poti vedea 2 Routere (R1 si R2), 1 PC si 1 Server. Conexiunea dintre PC si Router, respectiv Server si Router formeaza (2) retele de tip LAN (Local Area Network). Conexiunea intre cele 2 Routere formeaza o retea WAN pentru ca R1 si R2 conecteaza mai multe retele (LAN-ul cu PC-ul si LAN-ul cu Serverul). Daca inca nu iti este foarte clar, nu-ti fa probleme pentru ca vom discuta mai in detaliu despre aceste concepte pe parcursul acestei carti.

O retea de tipul **WLAN** este o retea LAN la care ne putem conecta wireless de pe telefonul mobil, tableta, laptop sau orice alt dispozitiv. Acel mediu wireless este unul separat fata de cel fizic si contine proprietati (viteza, securitate, raza de acoperire, etc.) diferite fata de acesta.

2) Topologii de retea

O retea (si dispozitivele acesteia) poate fi reprezentata in mai multe moduri, fiecare avand un scop diferit. De exemplu, daca suntem interesati doar de conectivitatea in retea (sau in Internet), atunci vom grupa dispozitivele intr-o **topologie de tip star (stea)** pe care o poti vedea in figura 1.2.

Singura problema cu acest tip de topologie este lipsa redundantei pentru fiecare dispozitiv din retea. De ce? Pentru ca fiecare dispozitiv (ex: laptop, PC, imprimanta etc.) va fi conectat printr-un singur cablu la retea. Daca se intampla ceva cu acel cablu (nu este mufat corect, este scos din greseala din Switch, este taiat din greseala), atunci conexiunea la retea (respectiv la Internet) va fi pierduta.

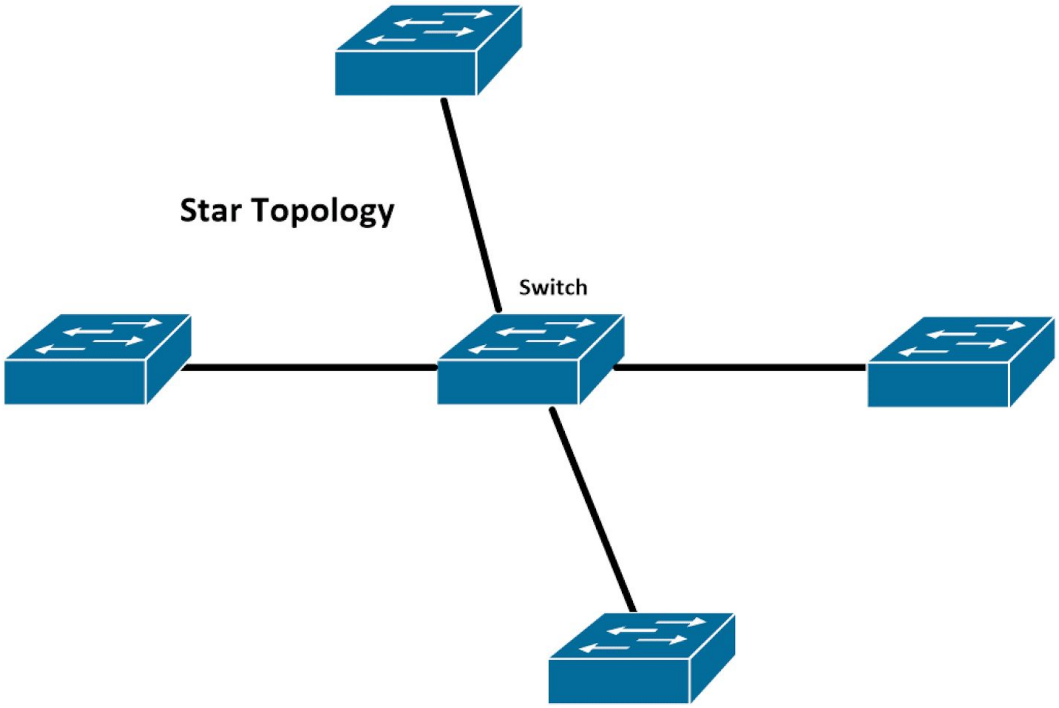


Figura 1.2

Avem o solutie pentru aceasta problema care este folosita mai des pe echipamentele de retea (Switch-uri, Routere, Firewall-uri etc.) pe care o putem ilustra printr-o **topologie** numita **full mesh** (“plasa” - de paianjen) pe care o poti vedea in figura 1.3.

Topologiile Full Mesh ne asigura redundanta in retea, datorita interconectarilor care exista intre echipamentele de retea. Astfel daca se intampla ceva cu un cablu sau cu un echipament doar o mica parte a retelei va fi afectata (si nu intreaga retea).

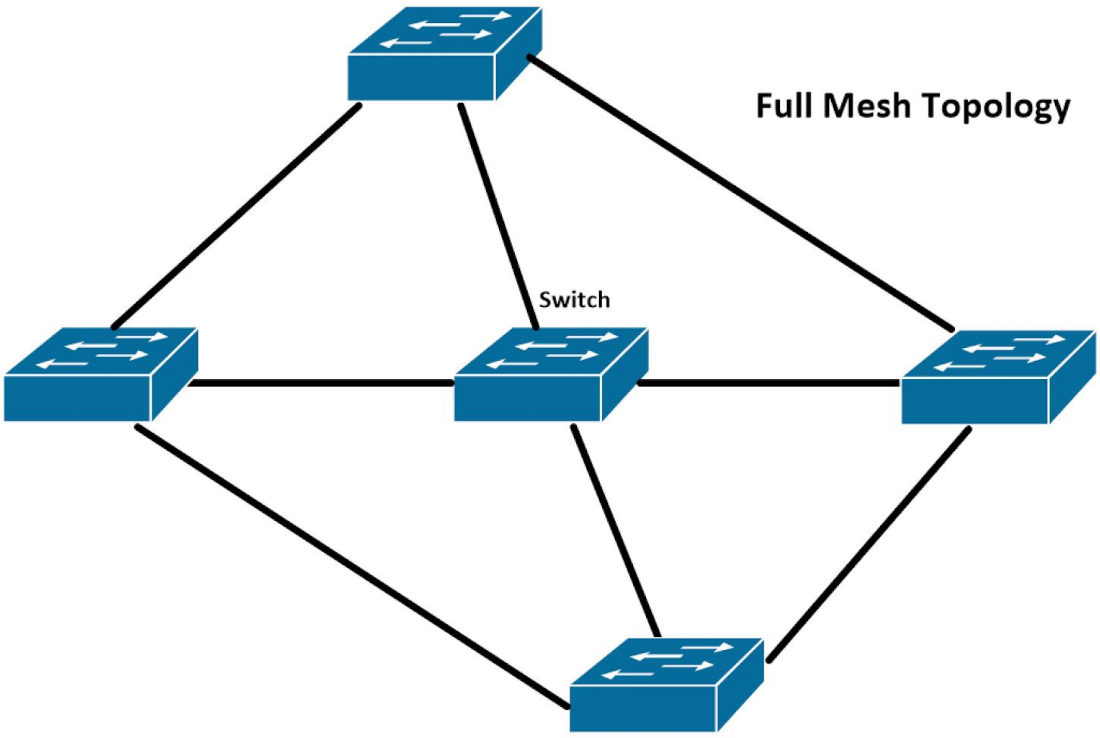


Figura 1.3

Desi suna bine si poate te intrebi "*de ce nu folosim mereu acest tip de topologie?*" vreau sa-ti spun ca lucrurile nu stau chiar asa. Am uitat sa luam in considerare un factor extrem de important cand vine vorba de retele de calculatoare pe care multa lume nu il mentioneaza: **Costul**. O topologie Full Mesh implica costuri mult mai ridicate fata de o topologie Star.

Ce fel de costuri? Echipamente mai bune (Switch-uri, Routere etc.) cu un numar mai ridicat de porturi, interfete mai multe pe PC-uri, laptop-uri si Servere, o infrastructura a cablurilor de retea mult mai buna (si totodata un numar de cabluri mult mai ridicat).

Astfel a fost gandita un altfel mod de reprezentare a retelelor (aka. topologie) care le imbina pe ambele: **Partial-Mesh**. Acest tip de topologie adopta un nivel de redundanta si de costuri mediu, astfel incat sa le imbine pe cele doua. Aceasta topologie o poti vedea in figura 1.4 si poti vedea cum imbina cele 2 elemente discutate mai devreme.

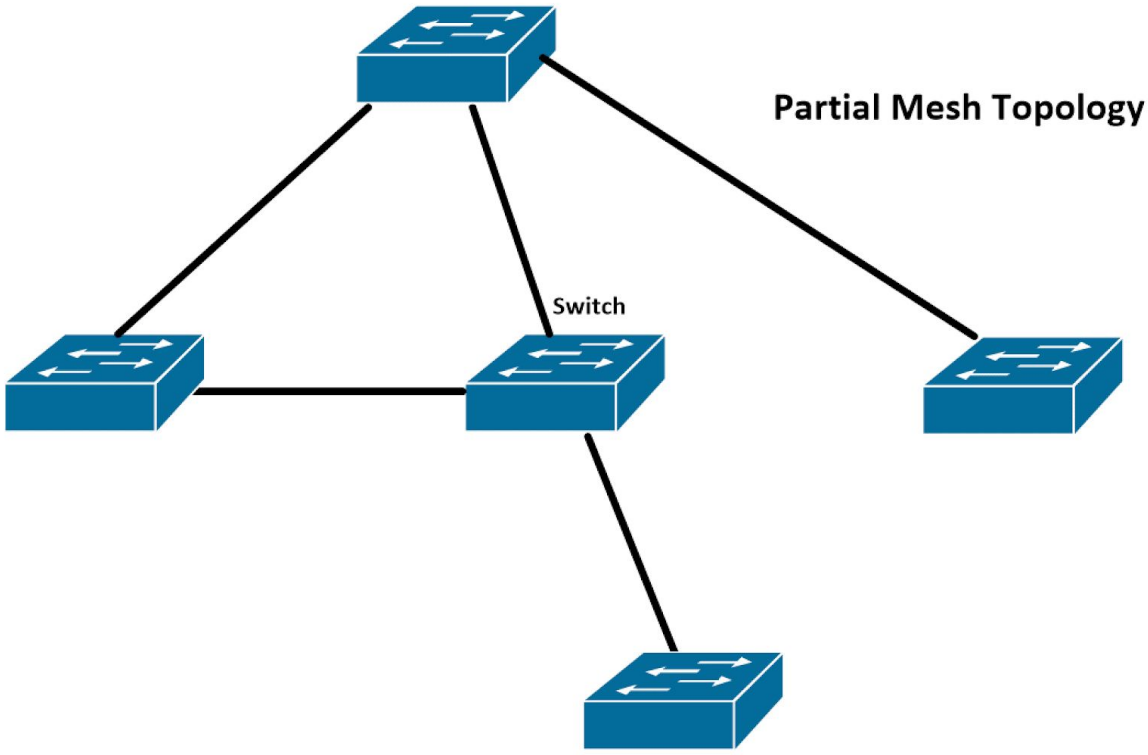


Figura 1.4

Acum, mergand mai departe mai vreau sa-ti prezint inca o topologie principala care se foloseste mai ales peste Internet, in retelele WAN. Acesta topologie se numeste **Hub and Spoke** si are la baza Routere. Scopul acestor Routere este sa lege retelele intre ele prin Internet (asa cum poti vedea in figura 1.5).

Dupa cum poti sa vezi, Routerul R2 este “Hub-ul” (ceea ce inseamna ca R1 si R3 vor sti de el si se vor conecta la acesta), iar R3, respectiv R1 sunt “Spoke-urile”.In acest scenariu, R2 este Routerul principal care va facilita conexiunea intre retelele Routerelor R1 si R3. Asta inseamna ca: pentru ca R1 sa ajunga la R3, trebuie sa trimita traficul catre R2 pentru ca acesta stie unde se afla destinatia R3. Acesta este un procedeu de rutare a pachetelor in Internet despre care vom discuta mai in detaliu in capitolul 9.

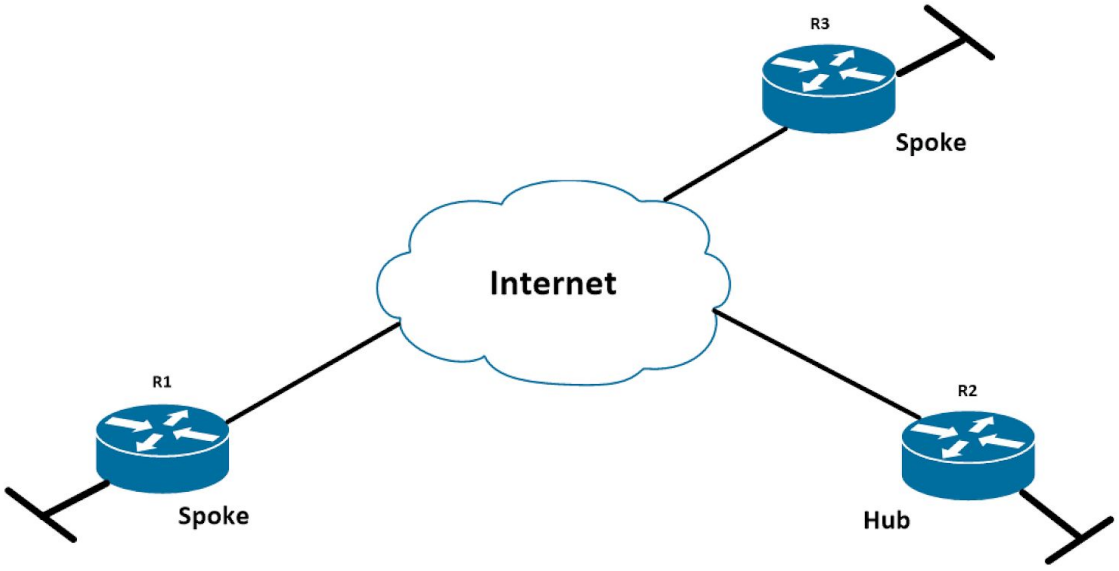


Figura 1.5

3) Componentele unei retele

O *retea este alcatuita* din:

- ***End-device*** (PC, Laptop, Smartphone, Servere etc.)
- ***Switch*** - interconecteaza mai multe end-device-uri intr-o retea
- ***Router*** - interconecteaza mai multe retele
- ***Firewall*** - *ne protejeaza reteaua de posbile atacuri din Internet*
- ***Mediu de transmisie*** – cablu (cupru), lumina (fibra optica), wireless (aer)

Acum, propun sa discutam despre cateva elemente de retea (din cele enumerate mai sus). Primele componente ale unei retele vor fi chiar end-device-urile:

A. END DEVICE-uri & MEDII DE TRANSMISIE

De obicei noi (consumatorii sau utilizatorii finali) suntem cei care folosesc dispozitivele terminale (end-device-uri). Fiecare dintre noi are un Laptop, un telefon sau o tableta cu care ne putem conecta la Internet. Ei bine, aceasta conexiune poate fi facuta folosind unul sau mai multe medii de transmisie a datelor (curent electric, impulsuri de lumina sau unde radio).

Cand ne conectam cu telefoanele noastre la Internet folosim conexiunea printr-un mediu wireless. Daca folosim un laptop sau PC putem opta pentru conexiunea prin cablu (UTP) sau prin wireless (folosind un adaptor USB sau o plata de retea speciala pe PC).

Figura 1.6

Fibra optica este folosita (de obicei) in momentul in care vrem sa conectam echipamente (de retea) intre ele: Switch - Switch, Server - Switch, etc. De ce ? Pentru ca fibra optica ofera suport pentru viteze de transfer mult mai ridicate (10, 40, 100 Gbps) pe o distanta mai lunga fata de UTP. Aceasta distanta poate fi intre 1 - 5 km (sau chiar mai mult), pe cand in cazul cablurilor UTP distanta maxima este de doar 100 de metrii.

De asemenea, fibra optica este tot mai des folosita in momentul de fata cand vine vorba de conexiunea utilizatorilor la Internet. Motivul principal se datoreaza vitezei de transfer ridicata suportata de catre fibra optica si distanta pe care aceasta o poate parcurge (fara intreruperi - echipamente intermediare).

B. SWITCH

Switch-ul este un echipament de retea care interconecteaza mai multe PC-uri (si nu numai: imprimante, telefoane IP, AP-uri, etc) la ACEEASI retea locala (LAN).

El este caracterizat de un numar mare de port-uri (in general 24 sau 48) toate fiind capabile de viteze intre 100 Mbps si 1 Gbps (sau chiar 10Gbps). Switch-ul foloseste adresele MAC. (vom vorbi mai in detaliu in Capitolul 2).

Iata o imagine cu un **Switch** profesional Cisco:

Figura 1.7

C. ROUTER

Un Router este un echipament de retea care are rolul de a interconecta mai multe retele (LAN) intr-o retea mai mare (WAN -Wide Area Network). Router-ul este dispozitivul care ne conecteaza la Internet. El se ocupa de trimiterea pachetelor, catre reteaua destinatie din Internet.

In comparatie cu Switch-ul, Router-ul are mult mai putine port-uri (intre 2 – 5) la viteze asemanatoarea (100 Mbps – 10Gbps, depinde de model). Dupa cum spuneam scopul acestui echipament este sa interconecteze retelele. Toate aceste retele au nevoie de un mod de identificare si astfel Routerele foloseste adresele IP (despre care vom vorbi in Capitolul 3).

Acesta este un **Router** profesional Cisco:

Figura 1.8

3) Cum reprezentam o retea prin Topologii Fizice si Logice

Reteaua o putem **reprezenta** printr-o **topologie**. Aceasta poate fi de 2 feluri: fizica sau logica. Topologia fizica descrie echipamentele si modul in care sunt acestea conectate, cablurile folosite.

Alt exemplu avand in componenta 2 Routere, 1 PC si 1 Server. Unul dintre Routere este conectat la Internet – ***Exemplu #1*** **Topologie Logica**

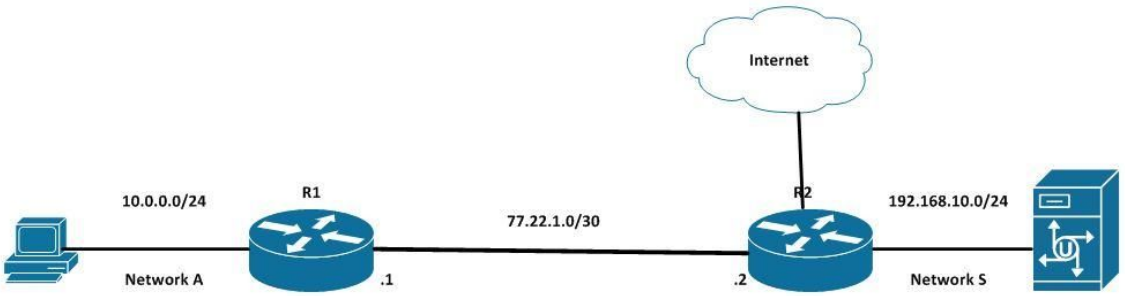


Figura 1.9

Aceasta retea contine 1 Switch-uri, 3 Routere, 2 PC-uri si 1 Server – ***Exemplu #2*** **Topologie Logica**

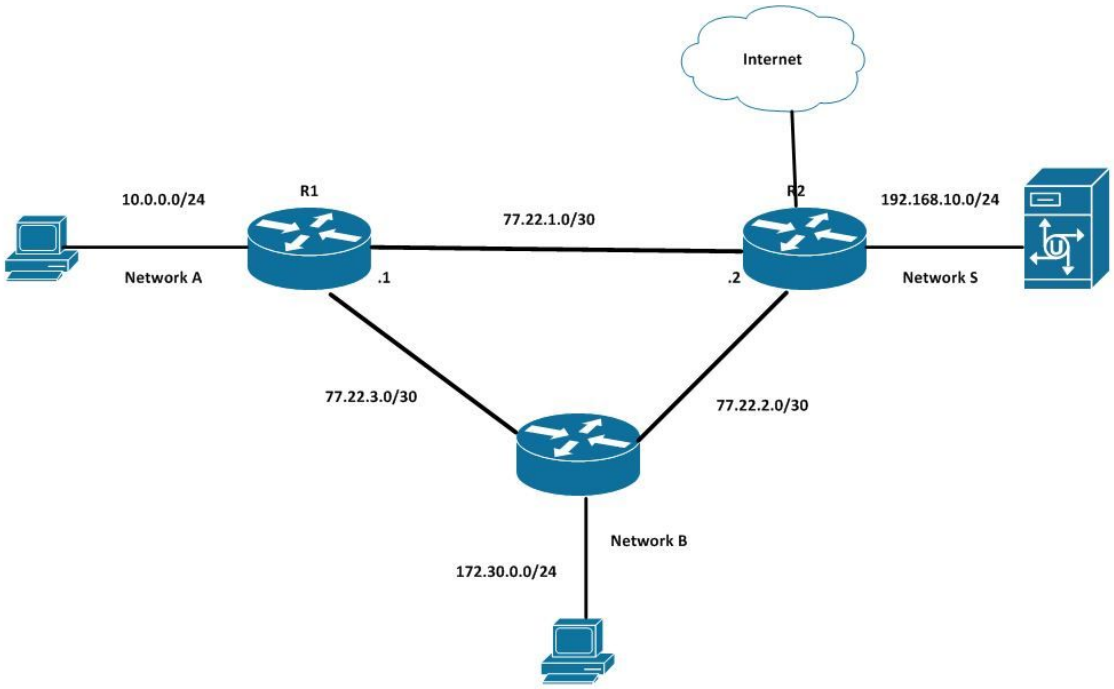


Figura 1.10

In exemplul Cisco, din figura de mai jos, putem vedea o reprezentare fizica (**Topologie Fizica**) a unei retele.

Ea defapt ne arata unde vor fi amplasate echipamentele si ce scop vor avea ele (conectarea la retea a PC-urilor unui departament, conectarea serverelor din Data Center la retea, conectarea punctelor de acces wireless la retea etc.)

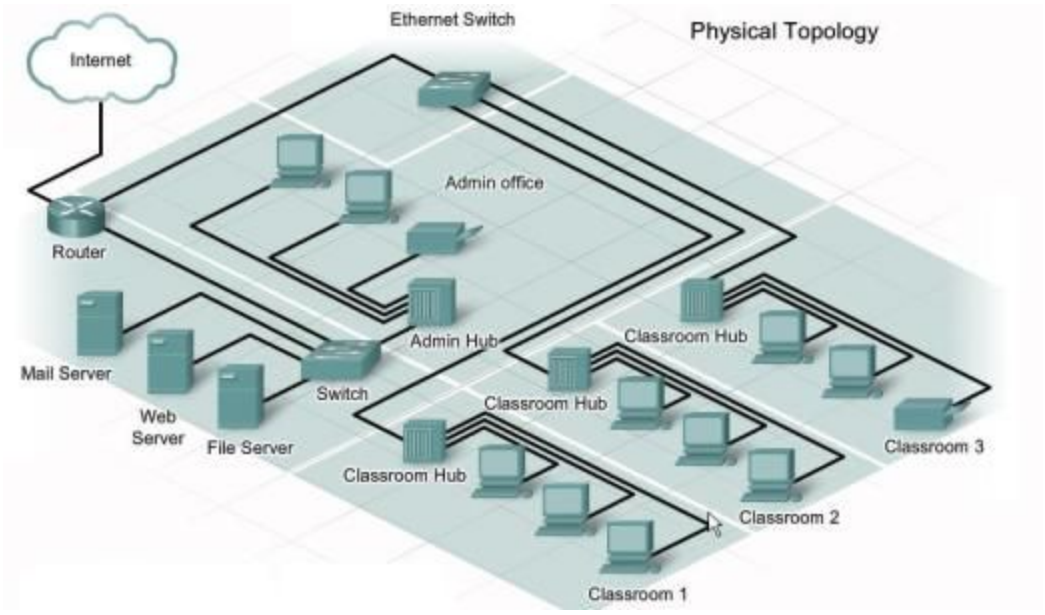


Figura 1.11

4) Cum comunica calculatoarele in retea ?

Pentru a putea comunica, dispozitivele (PC-uri, routere, switch-uri, etc) trebuie sa aiba un identificator unic. In acest caz este vorba de **IP (Internet Protocol)**.

IP-ul este modul prin care identificam un dispozitiv intr-o retea. El trebuie sa fie unic. Nu pot exista 2 IP-uri la fel in aceeasi retea, deoarece va aparea un conflict la nivelul acesteia.

Hai sa ne gandim la IP ca la un CNP pentru device-uri. Ce rol are CNP-ul ? De a identifica in, mod unic, fiecare persoana din Romania. Statul ne identifica prin CNP (aka IP), iar oamenii ne identifica prin Nume sau Prenume (aka. adresa MAC).

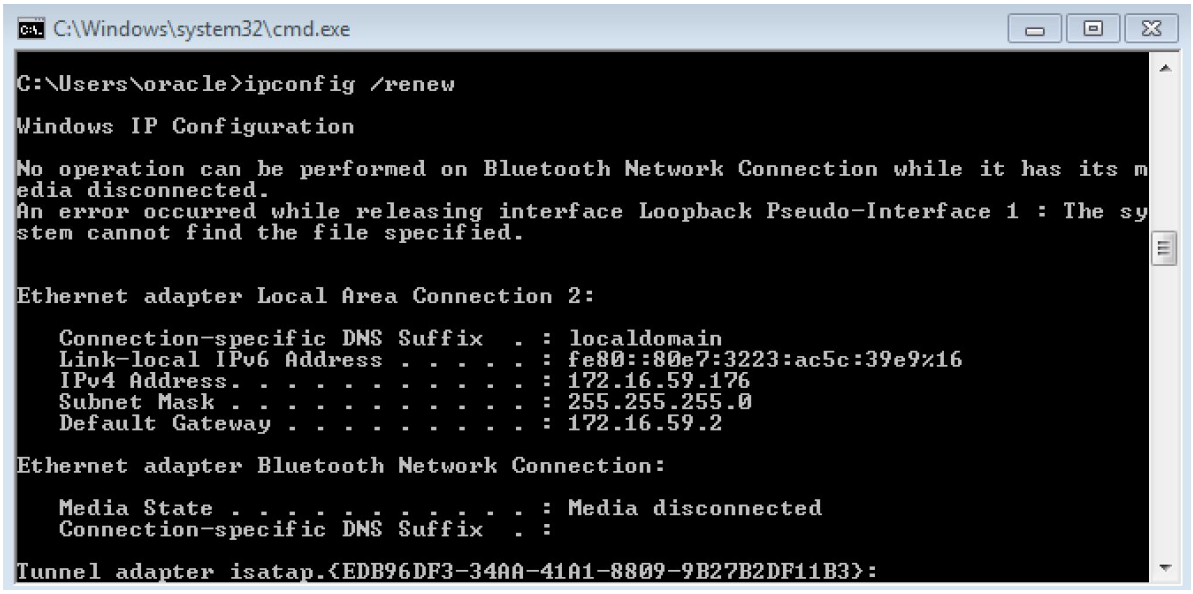


Figura 1.12

Exemplu de mai sus este luat din linia de comanda din Windows (**cmd**).

Aceste este *alt exemplu de IP*:

10.0.0.1/24, unde /24 reprezinta masca de retea,

Masca de retea (Subnet Mask) determina dimensiunea retelei (adica cate dispozitive se pot afla in aceeasi retea la un moment dat – 14 - (/28), 126 - (/25), 254 - (/24), 510 - (/23) etc).

Elementele necesare unui end-device pentru a comunica cu succes in Internet:

IP-ul = identifica, in mod unic, un dispozitiv conectat intr-o retea

Masca de Retea = determina dimensiunea retelei (ca numar de IP-uri disponibile)

Default Gateway = calea de iesire din retea (de obicei spre Internet printr-un Router)

Server DNS = "transforma" un nume (precum google.ro) intr-un IP (ex: 173.23.85.91)

Capitolul 2 - Modelul OSI

Modelul OSI este un standard (framework) care defineste modul de comunicare al echipamentelor dintr-o retea.

Acest model este impartit in **7 nivele**, fiecare independent de celalalt. Astfel, in timp pot aparea modificari (protocoale noi, imbunatatirea performantelor, la fiecare nivel in parte fara a influenta celelalte nivele). Standardul OSI are in componenta 7 nivele (il vei regasi si in figura 2.1):

1. **Physical**
2. **Data Link**
3. **Network**
4. **Transport**
5. **Session**
6. **Presentation**
7. **Application**

El ne ajuta (la inceput) extrem de mult la intelegerea conceptului de retele de calculatoare, iar pe masura ce prindem experienta la partea de **Troubleshooting** (rezolvarea problemelor care pot aparea intr-o retea, indiferent de dimensiunea acesteia).

Fiecare nivel in parte defineste cum trebuie sa decurga lucrurile in retea si in Internet. Toate au protocole care definesc comportamentul dispozitivelor din retea.

Spre exemplu: in momentul in care dorim sa trimitem trafic catre un anumit server vom folosi un protocol numit IP (acesta identifica in mod unic dispozitivile dintr-o retea sau din Internet). Sau poate dorim sa downloadam un fisier sau sa accesam o pagina web, atunci vom folosi protocoale precum FTP sau HTTP.

Iar acum, poate te intrebi: “*ce este un protocol ?*” Ei bine, e simplu:

Un **protocol** reprezinta un **set de reguli**. Atat ! un set de reguli pentru modul in care sa se comporte dispozitivle dintr-o retea.

De asemenea, ele sunt independente unul fata de celalalt, astfel daca apar anumite modificari, acestea nu ar trebui sa afecteze modul de functionarea al celorlalte.

Iata, in figura 2.1 de mai jos iti poti face o idee despre cum arata acest model:

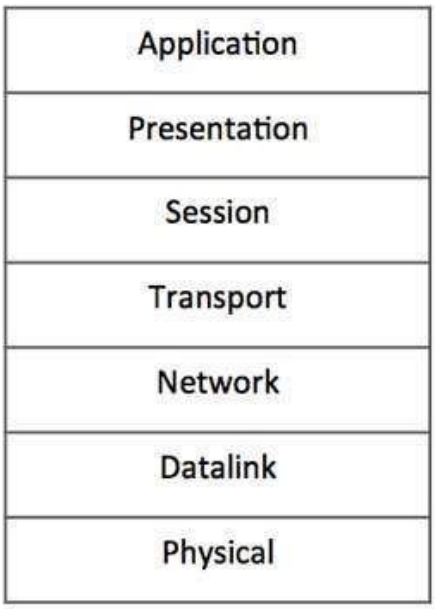


Figura 2.1

Dupa cum poti vedea in figura 2.2, fiecare nivel in parte este insotit (in partea stanga) de cateva o caracteristica, numita **PDU** (**P**rotocol **D**ata **U**nit) - sau unitatile de date pe care fiecare nivel in parte le foloseste:

- *Layer 1* foloseste **Bits**
- *Layer 2* foloseste **Frame-uri**
- *Layer 3* foloseste **Pachete**
- *Layer 4* foloseste **Segmente** (sau **Datagrame**)
- *Layer 5* foloseste **Date**
- *Layer 6* foloseste **Date**
- *Layer 7* foloseste **Date**

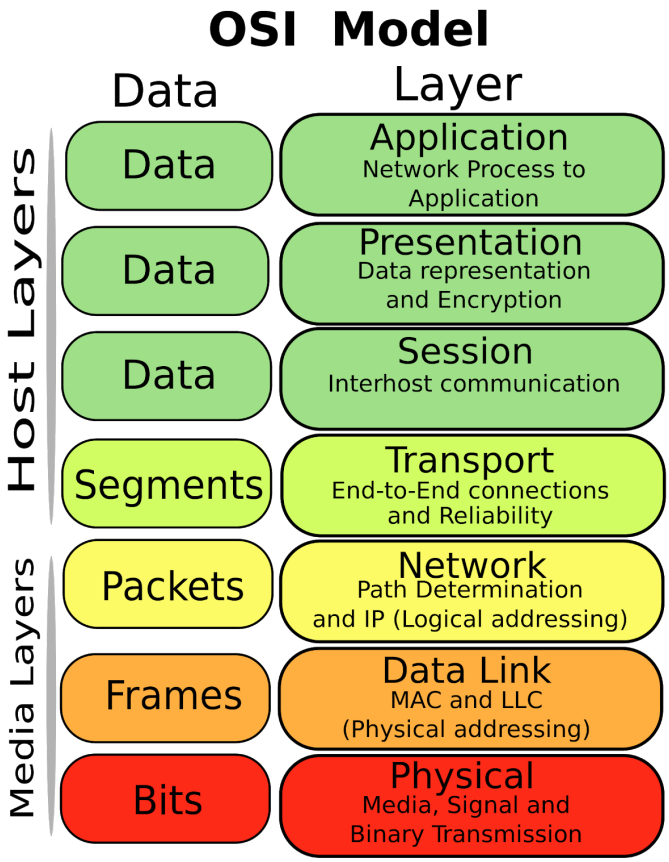


Figura 2.2

Vom vorbi mai in detaliu despre fiecare element in parte in capitolele dedicate fiecarui nivel. In “competitie” cu acest standard se afla **TCP/IP**, care a fost creat si adoptat mai repede. In prezent modelul **TCP/IP este folosit** (vezi figura 2.3). Acesta contine doar **4 nivele** (spre deosebire de cele 7 ale OSI).

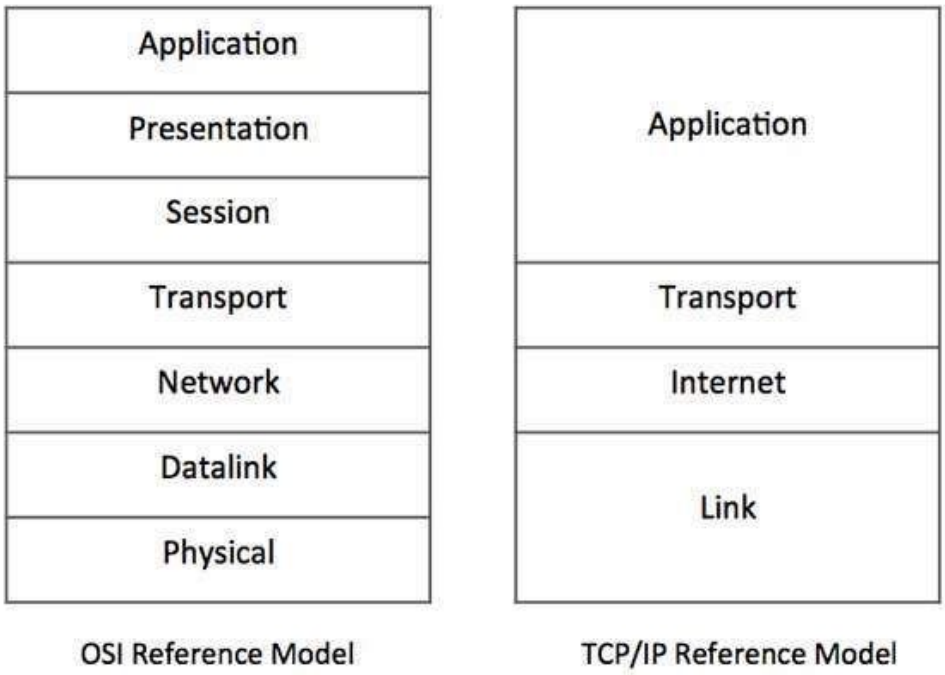


Figura 2.3

Acest model TCP/IP, in primul sau nivel inglobeaza cele 2 nivele (Physical si Data-Link) care fac parte din modelul OSI, iar ultimele 3 nivele (Application, Presentation si Session) sunt reprezentate ca un singur layer in TCP/IP si anume ca Application,

Scopul acestor modele este sa permita comunicatia dintre 2 device-uri (pe care sunt existente diferite aplicatii). De la **Nivelul Aplicatie** la **Nivelul Fizic** are loc un proces care se numeste **encapsulare**.

Adica un fisier (ex: o poza) *este sparta in bucatele* si se adauga informatii de la fiecare nivel in parte, dupa care este trimisa (sub forma de biti) catre destinatie, unde are loc **procesul invers** (Nivelul Fizic -> Aplicatie) numit **decapsulare**.

Fluxul de date si tot acest proces va arata in felul urmator (figura 2.4):

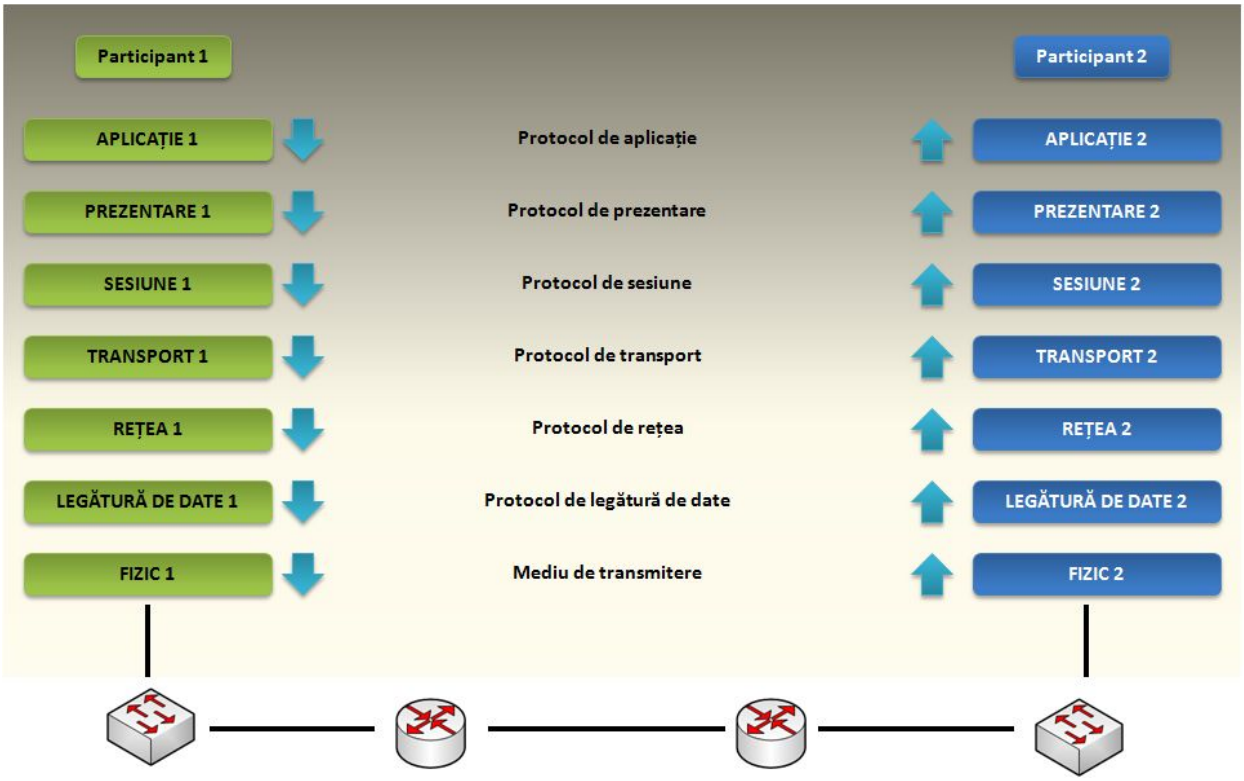

Participant 1
Participant 2
APLICAȚIE 1
Protocol de aplicație
APLICAȚIE 2
PREZENTARE 1
Protocol de prezentare
PREZENTARE 2
SESIUNE 1
Protocol de sesiune
SESIUNE 2
TRANSPORT 1
Protocol de transport
TRANSPORT 2
REȚEA 1
Protocol de rețea
REȚEA 2
LEGĂTURĂ DE DATE 1
Protocol de legătură de date
LEGĂTURĂ DE DATE 2
FIZIC 1
Mediu de transmitere
FIZIC 2

Figura 2.4

Capitolul 3 - Nivelul 1 - Fizic

Cand vorbim de nivelul fizic cel mai des ne referim la mediul prin care transmitem informatia. Cele 3 moduri principale prin care putem conecta retele (si dispozitive la retele) sunt:

1. Curent electric (cablu UTP cu 8 firicele de cupru)
2. Unde de lumina (fibra optica)
3. Unde radio (mediul wireless)

Acest nivel sta la baza retelelor de calculatoare pentru ca ne ofera conectivitatea fizica intre retele (locatii).

Fiecare dintre medile de transmisie au avantajele si dezavantajele lor. De exemplu: avantajul pentru **mediul wireless**, fata de celelalte 2, este evident (mobilitatea, flexibilitatea), dar contine si dezavantaje (securitatea, viteza mai mica).

Conexiunea prin Cablu (de obicei UTP) este mult sigura si de incredere fata de cea wireless. Ea poate transporta datele la o **viteza mult mai mare** (1/10/40 Gbps) si poate fi folosita pe distante mai mari (intre maxim 80 - 100 m).

Conexiunea prin **Fibra Optica** ne permite sa transportam date cu o viteza superioara pe distanta mult mai lungi (1 - 5 km sau chiar mai mult) fata de conexiunea prin cablu UTP.

Tipuri de cabluri UTP

Cand vine vorba de conectarea a 2 dispozitive (PC la Laptop sau Router la Laptop, Switch la Switch, etc.) lucrurile par foarte simple. Pur si simplu iei un cablu si le conectezi, si totul merge perfect, nu ? Ei bine nu e chiar asa.

Un cablu **Ethernet UTP** (**U**nshielded **T**wisted **P**air) contine **8 fire/pini** (4 folositi pentru Send si 4 pentru Receive) si arata cam asa (figura 3.1):

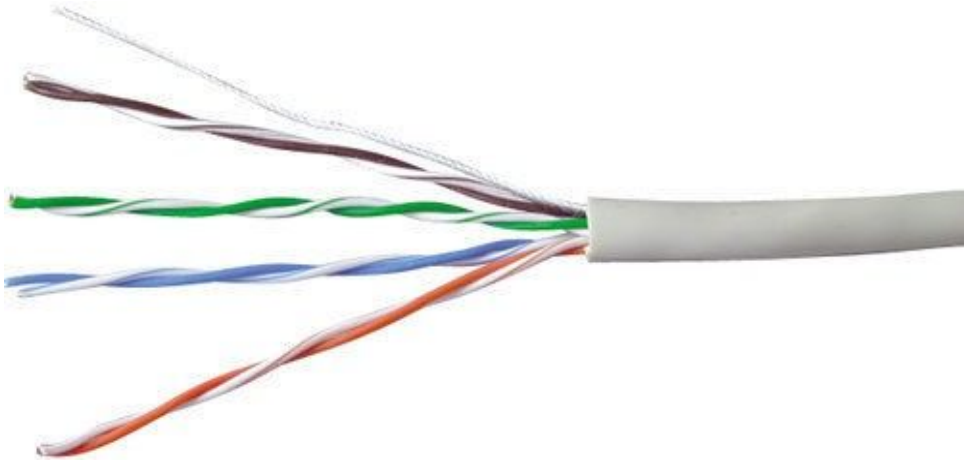


Figura 3.1

Acest cablu are 2 astfel de capete care pot fi “mufate” in moduri diferite. Aceste moduri sunt:

- ***Straight*** - firele sunt la fel la ambele capete
- ***Crossover*** - firele sunt incrucisate
- ***Rollover*** - folosit la consola, firele sunt date peste cap

Urmatoarele dispozitive folosesc un cablu **Straight**:

- Router <-> Switch
- Switch <-> PC
- Switch <-> Imprimanta

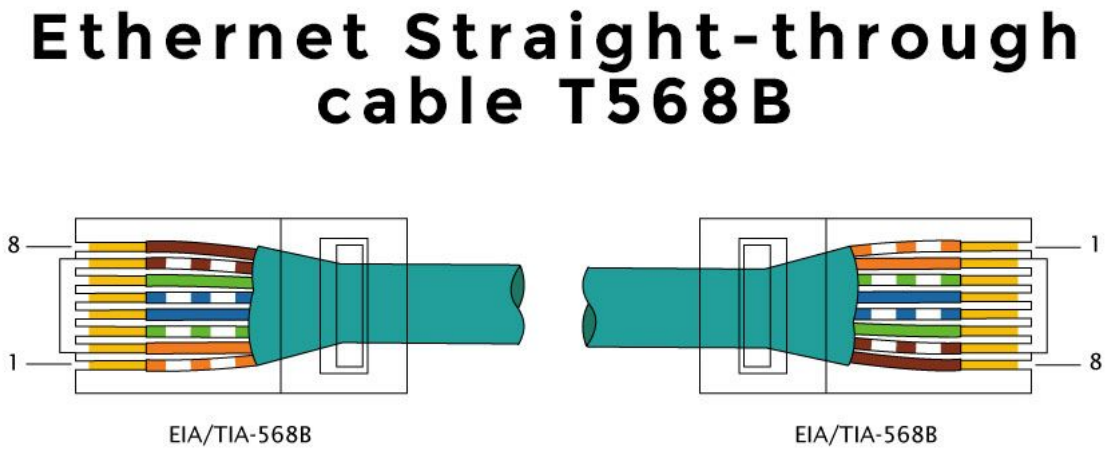


Figura 3.2

Urmatoarele dispozitive folosesc un cablu **Crossover**:

- Switch <-> Switch
- Router <-> Router
- Router <-> PC
- PC <-> PC

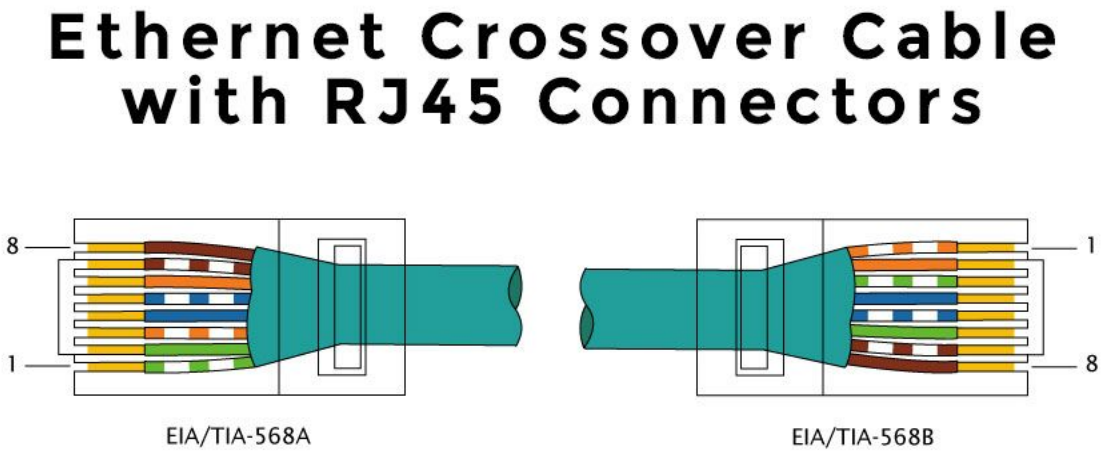


Figura 3.3

Cablul de Consola

Cand vine vorba de accesare unui Router avem 2 optiuni:

- Direct conectati la **consola** acestuia
- **Conexiune** de la **distanta** prin retea (**Telnet** sau **SSH**) - despre care vom vorbi mai tarziu in capitolul 7

Daca dorim sa ne conectam la echipament prin consola trebuie sa avem **acces fizic** la acesta. In cele mai multe cazuri acest lucru nu este posibil. La inceput cand configuram *“de la 0”* un dispozitiv de retea, ne vom lega prin ***consola*** la acesta, deoarece el nu are un IP la care sa ne putem conecta de la distanta.

Nevoia de a folosi consola mai apare si cand pierdem accesul la dispozitiv (s-a intamplat ceva cu reteaua sau cu echipamentul in sine) deoarece atunci putem investiga incidentul.

Pentru a ne conecta la orice echipament Cisco prin consola avem nevoie de un cablu special, numit **rollover** (vezi figura 3.4). Acest tip de cablu va fi introdus in **portul “de Consola”** al dispozitivului.

Figura 3.4

De asemenea, vom avea nevoie de un program (PuTTY - in figura 3.5 de mai jos, TeraTerm, SecureCRT, etc.) cu care sa ne putem conecta direct la Router.

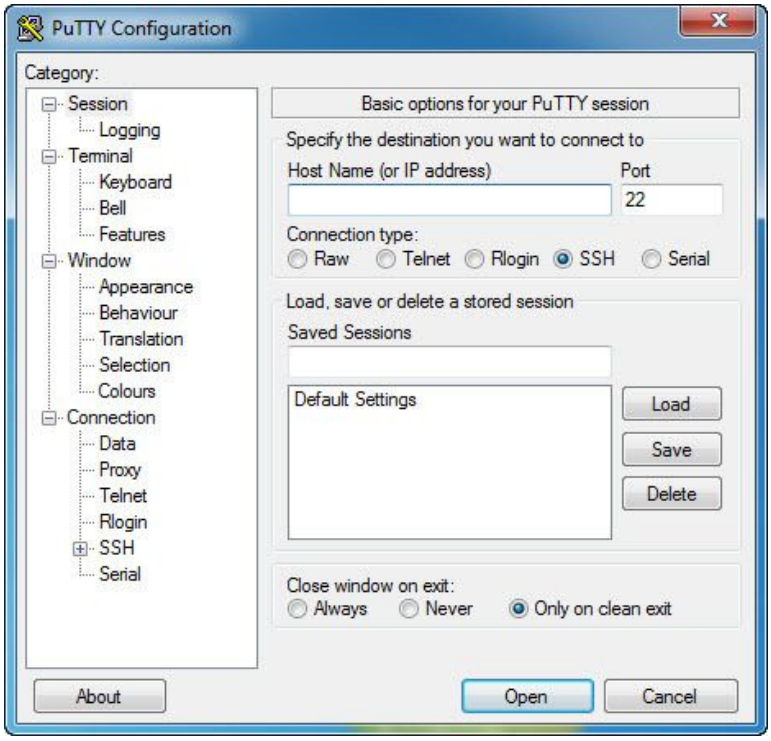


Figura 3.5

Porturi / Interfete si viteza acestora

Un port reprezinta modul in care ne putem conecta noi la retea (practic, reprezinta **partea fizica** si este “in care intra intra cablul"). O **interfata** este partea "**logica**" (virtuala) a portului (pe aceasta putem seta o adresa IP).

- *In port introducem cablul fizic*
- *Pe interfata setam adresa IP*

Pentru fiecare port in parte, viteza poate fi diferita. Aceasta poate varia intre 10 Mbps - 100 Gbps (100.000 Mbps), depinde de model si caz.

Unde:

Mbps = Mega biti pe secunda

Gbps = Giga biti pe secunda

In prezent, vom intalni cel mai des viteze de 1000 Mbps (1 Gbps). In figura 3.6, poti vedea un Switch Cisco cu 48 de porturi GigabitEthernet (adica cu viteza de 1000 Mbps) si 4 porturi (in drepata) 10 GigabitEthernet (adica 10.000 Mbps sau 10 Gbps).

Figura 3.6

Pentru Switch-uri este normal sa aiba atat de multe porturi (52 in acest caz, dar exista si modele cu sute de porturi) pentru ca acest echipament are rolul de a conecta mai multe dispozitive in aceeasi retea. In capitolul urmator vom vorbi mult mai in detaliu despre Switch-uri si cum functioneaza ele.

Full Duplex / Half-Duplex

Un alt element important cand vine vorba de modul de functionare al unui port este cel de cum se trimit defapt datele.

1. Un Switch (si un PC) poate trimite datele atat pe rand (PC-ul trimite, Switch-ul doar primeste si invers: cand Switch-ul trimite, PC-ul doar primeste) - acest mod se numeste **Half-Duplex** pentru ca orice dispozitiv poate doar sa trimita sau sa primeasca, dar NU in acelasi timp (figura 3.7).

2. Un Switch (si un PC) trimit si primesc datele simultan - acest mod se numeste **Full-Duplex** si este cel folosit (figura 3.8).

Beneficiul folosirii **Full-Duplex** este clar**: Viteza mult mai mare in retea.**

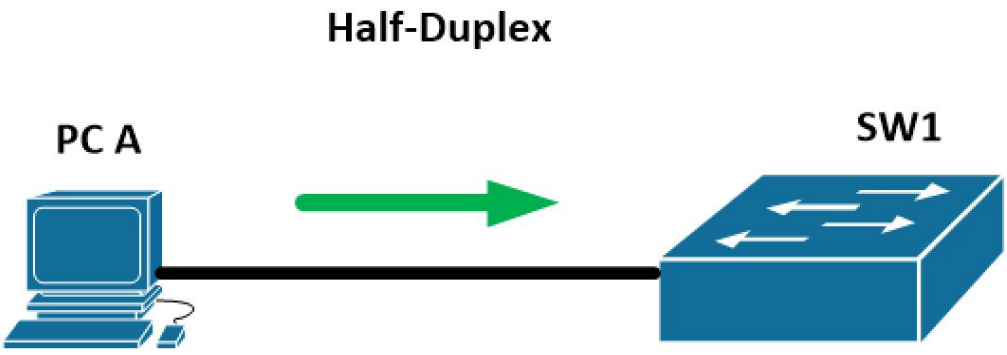


Figura 3.7

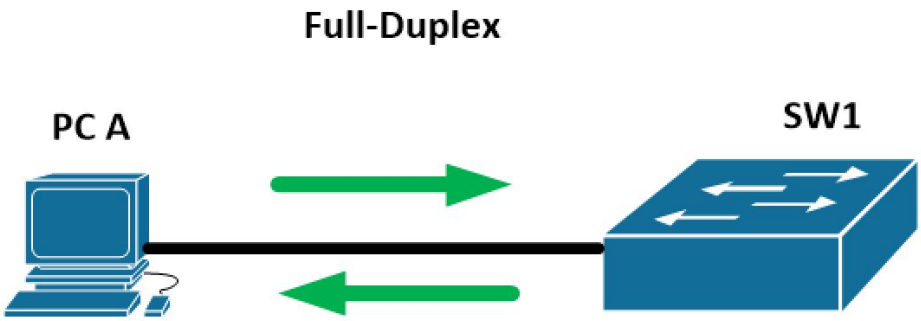


Figura 3.8

Domenii de coliziune. Domenii de broadcast

In orice retea, un pachet poate atinge diferite puncte din retea sau poate intra in conflict cu alte pachete transmise in mod simultan: aici intervin aceste domenii de broadcast sau de coliziune.

1) Domeniul de coliziune

Coliziunea se refera la faptul ca 2 device-uri diferite trimit un pachet in retea in acelasi timp. Daca cele 2 pachete sunt trimise simultan, atunci se formeaza o coliziune (pentru ca la inceputul tehnologiei de networking device-urile comunicau doar in mediu half-duplex - un singur device din retea trimitea datele, iar celelalte trebuiau sa astepta ca acele date sa fie receptionate in totalitate pana cand un alt device sa poate trimite alte date). Acesta era modul in care functionau retelele in anii ‘90 si inceputul anilor 2000. Mecanismul de rezolvarea a coliziunilor se numea CSMA/CD (Carrier Sense Multi Access with Collision Detection).

O **coliziune** *poate avea loc* doar pe un **segment** (legatura dintre 2 dispozitive: Switch - PC, PC - PC, Switch - Router, etc.) **half-duplex**, dintr-o retea. Astfel de coliziuni nu mai au loc (sau au loc foarte rar datorita unor erori) pentru ca fiecare echipament transmite traficul in modul **full-duplex**.

2) Domeniul de broadcast

Un domeniu de broadcast reprezinta distanta pe care un pachet (de tip broadcast) o face prin retea. Altfel spus: cat de departe poate ajunge un pachet broadcast va reprezenta domeniul de coliziune.

In figura 3.10 poate te vei intreba de ce legatura dintre Routere reprezinta un domeniu de broadcast ? Nu e cumva un domeniu de coliziune ?

Raspunsul este da si da :) este un domeniu de broadcast pentru ca **fiecare interfata a unui Router este o alta retea** si este un domeniu de coliziune pentru ca este un segment de retea si dupa cum am spus si mai devreme: o coliziune poate avea loc pe orice segment de retea (de obicei half-duplex).

Imaginile de mai jos (figura 3.9 si 3.10) reprezinta domeniile de coliziune, respectiv domeniile de broadcast dintr-o retea. Dupa cum poti vedea in figura 3.9 exista **4 domenii de coliziune**:

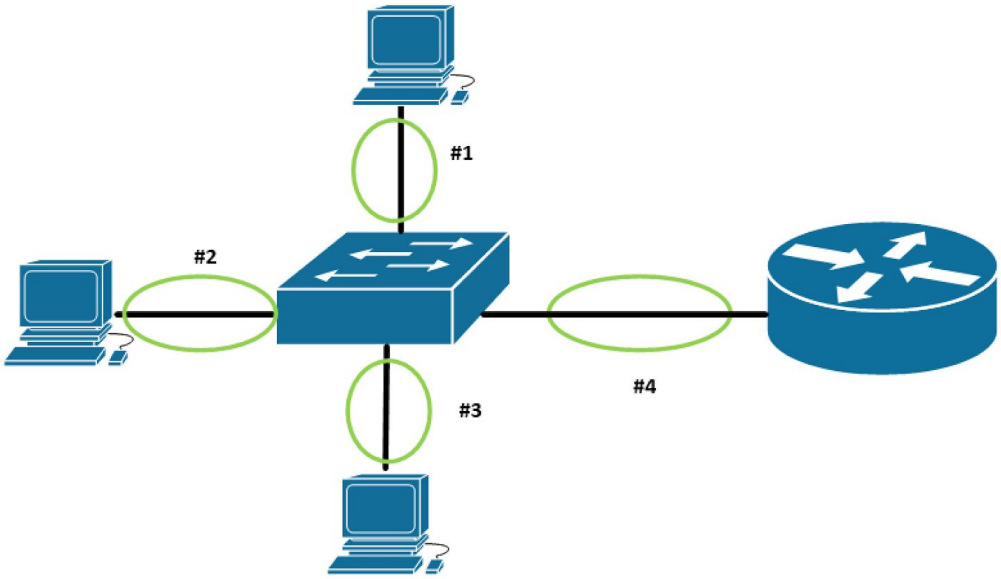


Figura 3.9

Iar in figura 3.10 exista **4 domenii de broadcast**:

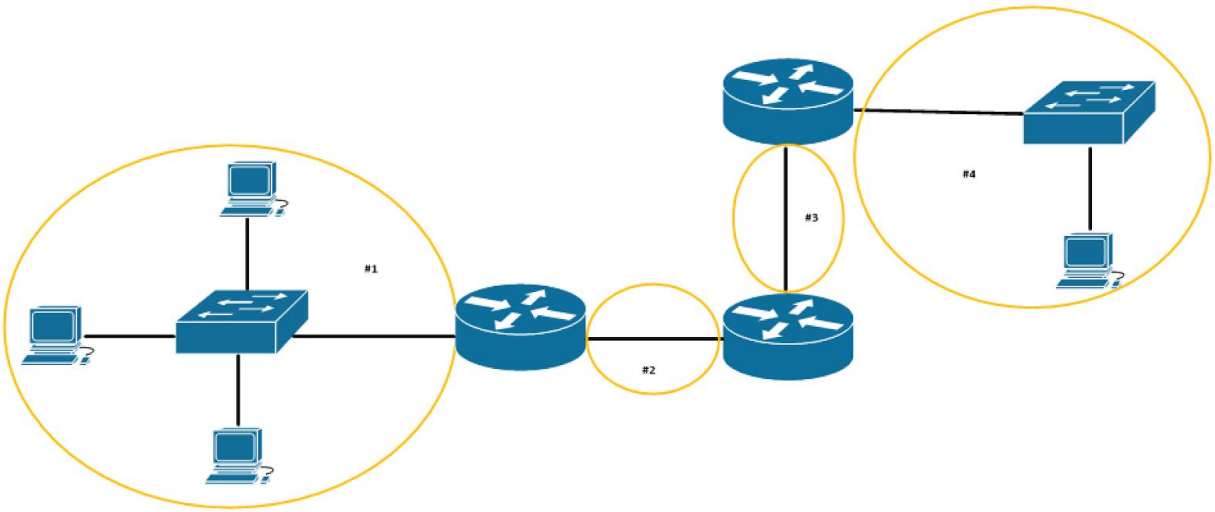


Figura 3.10

Capitolul 4 - Nivelul 2 - Legatura de Date (Data-Link)

Concepte de Baza despre Switch

Switch-ul este un echipament de retea care functioneaza la **nivelul 2 al modelului OSI**. Scopul acestuia este de a conecta mai multe device-uri (PC-uri, Laptop-uri, Servere, Imprimante, etc.) in aceeasi retea locala (LAN). Acest echipament contine mai multe **port-uri (24 sau 48**, depinde de model**)** care ii permit sa faca legatura in retea.

Figura 4.1

Marea majoritate a Switch-urilor din ziua de astazi folosesc **tehnologia Ethernet**. Unul din motive fiind acela ca ne ofera **viteze net superioare** (1 Gbps / 10 Gbps / 40 Gbps sau chiar 100 Gbps) fata de tehnologiile existente (Token Ring, FDDI, etc.) pe piata in momentul adoptiei.

Switch-ul trimite datele de la un dispozitiv la celalalt pe **baza adreselor MAC** (sursa si destinatie). O **adresa MAC** reprezinta un *mod de identificare unic*, al fiecarui device dintr-o retea. Acesta este scrisa, din fabrica, pe placa de retea a fiecarui PC, smartphone, laptop, tableta si are urmatoarea forma:

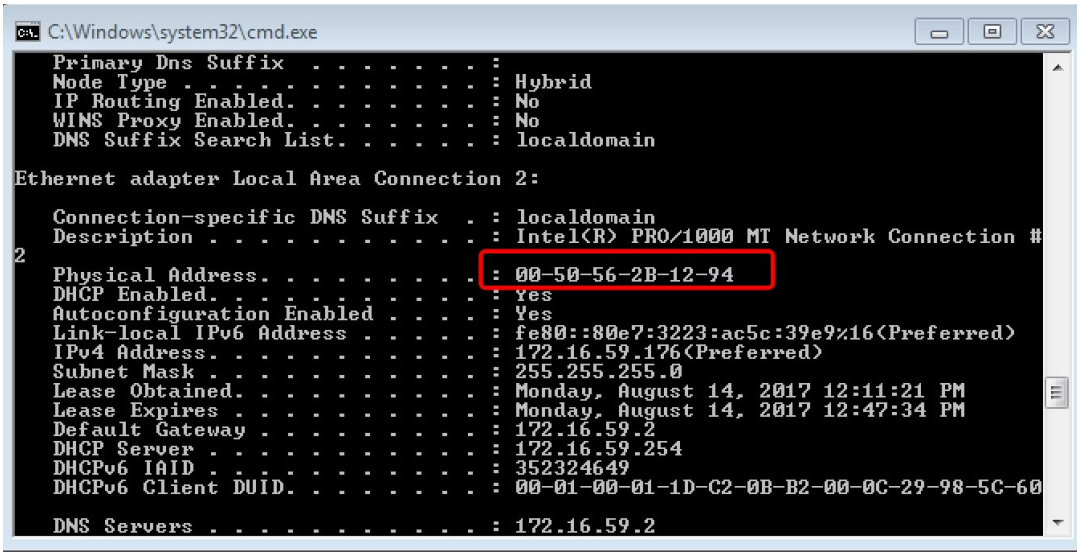


Figura 4.2

Este reprezentata in format hexadecimal, pe **48 de biti** (12 caractere, 4 biti fiecare caracter). Primii 24 de biti (00-1B-63) reprezinta partea specifica vendor-ului (ex: Cisco, Apple, Intel etc.), iar urmatorii 24 de biti (84-45-E6) reprezinta *partea specifica device-ului*, care il identifica in mod unic in retea.

Ce este Ethernet ?

De aceasta tehnologie sunt absolut sigur ca ai mai auzit pentru ca: 1. am amintit putin mai devreme de ea si 2. pentru ca Routerul tau de acasa foloseste o astfel de tehnologie (la fel si PC-ul / Laptop-ul - interfata de retea).

Ethernet este cea mai raspandita tehnologie (pe echipamente de retea) din ziua de astazi datorita acestor 2 lucruri:

- **viteza crescute** (10/40/100 Gbps)
- **forma de adresare** (adresele **MAC**)

Fiecare port al Switch-urilor sau al Router-elor este construit pe tehnologia Ethernet si este notat diferit, in functie de viteza :

- **FastEthernet** - 100Mbps (aka. Fa0/1 ... Fa0/24)
- **GigabitEthernet** - 1000Mbps (aka Gi0/1 ... Gi0/24)

Astfel pe masura ce device-urile trimit trafic in retea, Switch-ul va invata adresele (MAC) lor sursa si le va asocia cu porturile de pe care provin (portul la care e conectat device-ul - ex: Fa0/1, Fa0/10 sau Gi0/4 etc.)

Iata in figura 4.3 poti vedea cum arata headerul Ethernet:

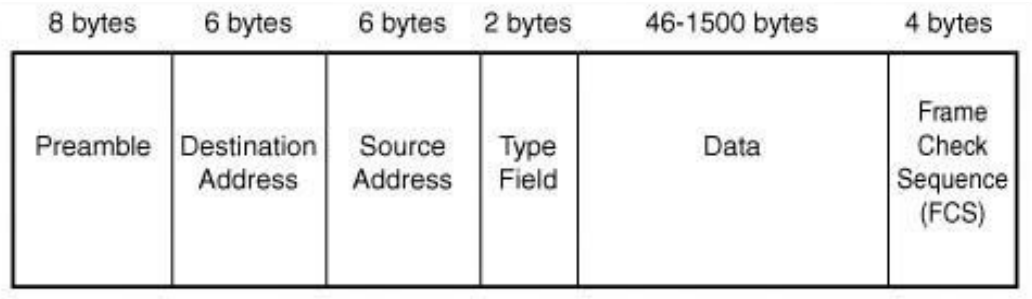


Figura 4.3

Structura headerului este in felul urmator:

1. **Preamble**
    - sir de biti care indica inceputul unui nou frame (“pachet”)

2. **Destination Address**
    - adresa MAC destinatie

3. **Source Address**
    - adresa MAC sursa

4. **Type Field**
    - Indica versiunea de Ethernet folosita si lungimea frameului

5. **Data**
    - reprezinta datele efective transmise (impreuna cu headerele nivelelor superioare)

6. **FCS**
    - mecanism de stabilire a integritatii pachetelor

7. **EoF** (End of Frame bits)
    - desi nu apare in imaginea de mai sus, exista un sir de biti speciali care indica terminarea frameului

Acum, hai sa vedem cum arata headerul Ethernet intr-un scenariu real din Wireshark (figura 4.4):

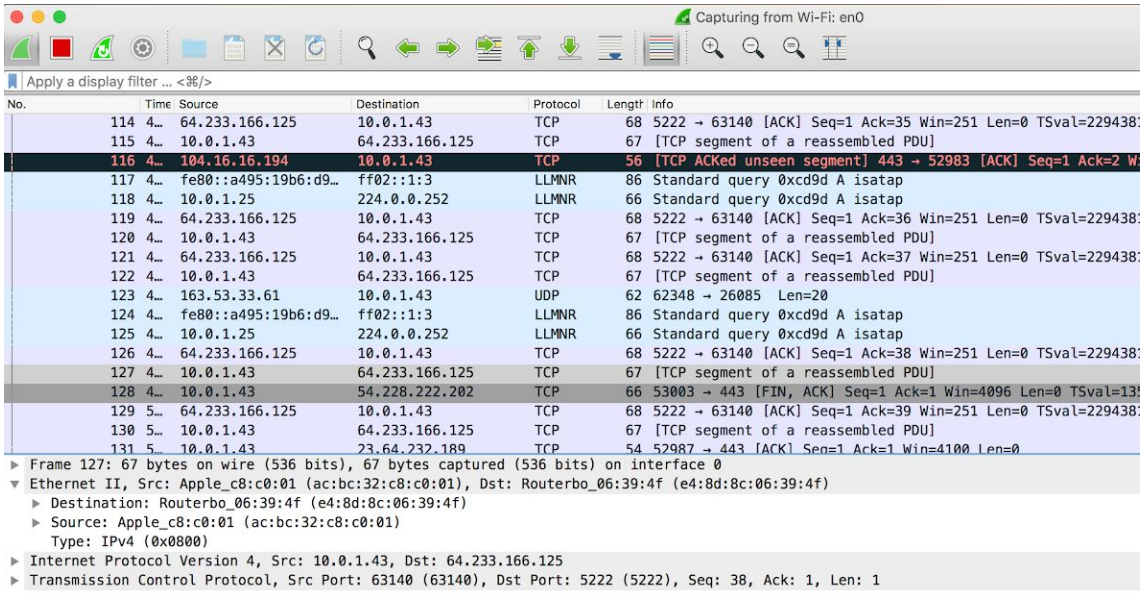


Figura 4.4

Dupa cum poti vedea in figura 4.4, un flux de date de tip TCP intre 2 device-uri care contine pe langa porturi, adrese IP (lucruri despre care vom discuta in capitolele urmatoare), mai contine si un camp (layer/nivel) special **Ethernet II**, unde poti identifica cu usurinta cele **2 adrese MAC** (*destinatie* si *sursa*).

Acum ca am vorbit despre ce inseamna standardul Ethernet si de ce este folosit propun sa discutam despre Switch-uri si sa vedem cum foloseste el acest standard (mai degraba cum foloseste adresele MAC pentru a facilitatea comunicarea in aceeasi retea - LAN).

Cum invata si foloseste un Switch adresele MAC ?

Scopul unui Switch este cel de a interconecta mai multe dispozitive in aceeasi retea locala (LAN). El face asta cu ajutorul adreselor MAC (sursa si destinatie). Adresa MAC destinatie este folosita pentru trimiterea traficului catre o destinatie anume, iar adresa MAC sursa este folosita pentru a memora portul pe care se afla un dispozitiv.

Deci practic, Switch-ul invata despre fiecare device din retea pe baza adresei MAC sursa si ia decizia de a trimite pe un port (spre destinatie) anume pe baza adresei MAC destinatie.

Switch-ul retine toate aceste **informatii** intr-o memorie speciala: **CAM** (**C**ontent-**A**ddressable **M**emory).

Aceasta tabela CAM descrie ce **adresa MAC** (sursa) se afla pe un **port** (altfel spus, face o mapare/asociere *adresa **MAC sursa** - **port***) - **exemplu**: Adresa MAC X se afla pe portul Fa0/5, Adresa MAC G se afla pe portul Fa0/9.

Hai sa luam reteaua din topologia din figura 4.5 in care apar 3 PC-uri care sunt legate la un Switch:

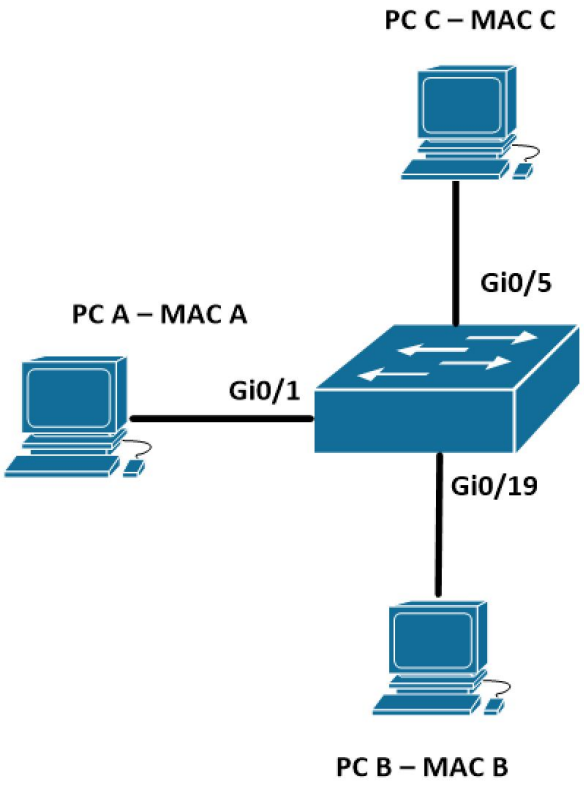


Figura 4.5

Practic o tabela CAM pentru Switch-ul din figura 4.5 va arata (la inceput) in felul urmator:

ID	Adresa MAC	Port
1		Gi0/1
2		Gi0/19
3		Gi0/5

Tabelul 4.1

Dupa cum poti vedea in Tabelul 4.1, la inceput CAM-ul este gol, Switch-ul nu stie de adresa MAC a niciunui dispozitiv din retea.

Acum sa ne imaginam ca cele 3 PC-uri comunica in retea (intr-un mod similar cu cei din pasii 1 - 5, de mai jos), iata un scenariu care va ilustra comportamentul acestora:

1) PC-ul A trimite un mesaj catre PC-ul B

Switch-ul invata adresa MAC a lui A.

ID	Adresa MAC	Port
1	A	Gi0/1
2		Gi0/19
3		Gi0/5

Switch-ul **trimite mesajul ca broadcast** pentru ca nu stie unde se afla PC-ul B.

2) PC-ul B primeste mesajul

ID	Adresa MAC	Port
1	A	Gi0/1
2		Gi0/19
3		Gi0/5

3) PC-ul B raspunde PC-ului A

Switch-ul **invata** adresa MAC a lui B si trimite, pe baza tabelei CAM, mesajul catre A.

ID	Adresa MAC	Port
1	A	Gi0/1
2	B	Gi0/19
3		Gi0/5

4) PC-ul B trimite un mesaj PC-ului C

Switch-ul **trimite mesajul ca broadcast** pentru ca nu stie unde se afla PC-ul C.

ID	Adresa MAC	Port
1	A	Gi0/1
2	B	Gi0/19
3		Gi0/5

5) PC-ul C raspunde PC-ului B

Switch-ul invata adresa MAC a lui C si trimite, pe baza tabelei CAM, mesajul catre B.

ID	Adresa MAC	Port
1	A	Gi0/1
2	B	Gi0/19
3	C	Gi0/5

Pe langa standardul Ethernet mai exista si alte standarde care functioneaza la acest nivel 2, dar despre care vom discuta cu alta ocazie:

- **PPP** (Point-to-Point Protocol)
- **PPPoE** (Point-to-Point Protocol over Ethernet) - folosit de ISP pentru autentificare
- **MPLS** - standardul curent de comunicare intre organizatii prin provideri (non-Internet)
- **Frame Relay / ATM** - tehnologii mai vechi care nu se mai folosesc

Capitolul 5 - Nivelul 3 - Retea (Network)

Concepte de Baza despre Router

Scopul unui Router este **de a interconecta mai multe retele LAN**, intr-o retea mai mare (adesea numita WAN). Tot ce trebuie sa faca acesta este sa ia urmatoarea decizie pentru fiecare pachet in parte:

"*Pe ce interfata trebuie sa trimit acest pachet ? Daca nu stiu unde sa-l trimit il voi arunca (drop).*"

Figura 5.1

ATENTIE! By default, **un Router cunoaste** doar retelele **Direct Conectate**. Acesta nu stie cum sa trimita mai departe de aceste retele, pachetele. Aici intervenim noi, cei care administram aceste echipamente si configuram rutele pe device.

In momentul in care porneste, un Router, invata mai intai de retelele direct conectate (cele care incep cu C, in tabelul de mai jos). In figura de mai jos poti vedea **Tabela de Rutare** a unui Router Cisco, care contine rutele/retelele direct conectate (C) si adresa IP a lui R1 de pe acele interfete (L).

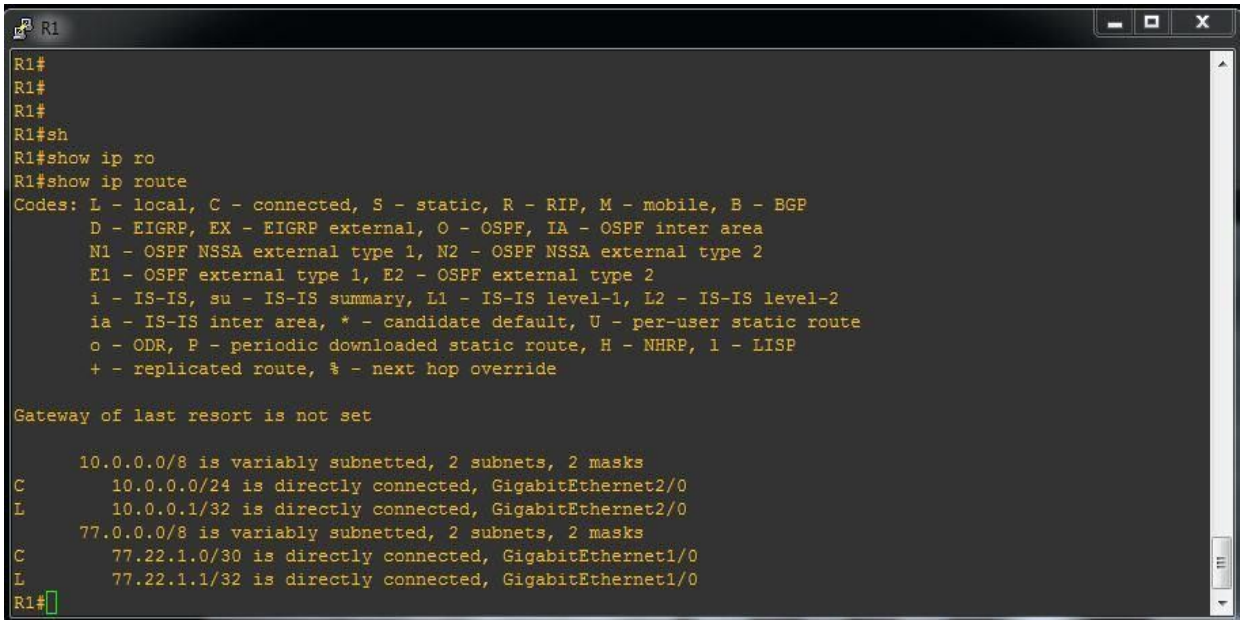


Figura 5.2

Pentru a putea face posibila toate acestea, Routerul foloseste adresele IP ca mod de referinta (*de la cine vine traficul* ? - **sursa** - si catre *cine trebuie sa trimit acest trafic* ? - **destinatia**).

Pentru a putea trimite traficul (pachetele) spre destinatie, Routerele trebuie sa cunoasca, in primul de rand, acele destinatii. Iar aici intervin mai multe moduri prin care un **Router poate invata anumite retele**:

- **Manual** - prin Rute Statice
- **Dinamic** - prin Protocoale de Rutare (RIP, OSPF, EIGRP)

In cele ce urmeaza vom incepe sa vorbim despre adresele IPv4 si IPv6, urmand ca in **capitolul 9**, sa trecem la **partea practica** si sa configuram un Router Cisco in simulatorul de retele de calculatoare Packet Tracer.

Ce este IPv4 ?

Protocolul IPv4 a fost dezvoltat in anii '80 si s-a propus folosirea a 32 de biti pentru definirea unei adrese (ex: 192.168.1.1). In fiecare camp din aceste 4 pot fi alocati 8 biti:

8 biti * **4** campuri = **32** biti.

Acum, hai sa ne gandim putin la acest numar de biti, 32. Acesta ne poate spune ceva legat de numarul maxim de adrese IP care pot fi generate: 2^32 ~= 4.2 Miliarde ! Da, ai citit bine, 4.2 miliarde de adrese IPv4... si **s-au terminat**.

TIP: de ce 2 ^32 ? deoarece fiecare bit poate lua valoare 0 sau 1, asadar daca avem 32 de biti vom putea genera aproximativ 4.2 miliarde de numere unice.

In anul 2011, mai exact in vara acelui an, **IANA (I**nternet **A**ssigned **N**umbers **A**uthority) a alocat ultimul spatiu de adrese IPv4. Asta inseamna ca nu mai putem conecta alte dispozitive la internet ? Nicidecum, de atunci si pana acum (2016) internetul a crescut foarte mult ca numar de dispozitive conectate. Iata urmatorul grafic:

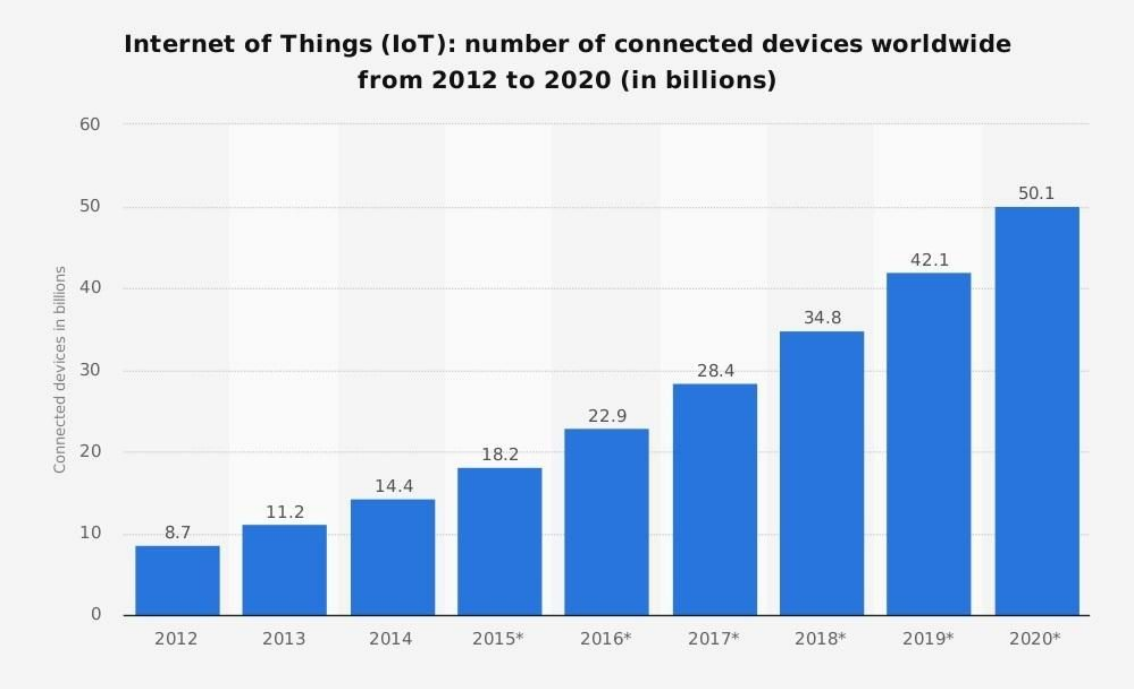


Figura 5.3

Dupa cum spuneam si mai devreme, numarul maxim de adrese IPv4 este de aproximativ ~**4.2 Miliarde**. In anul 2016 se estimeaza ca numarul total de dispozitive conectate la Internet este in jur de ~30 Miliarde, numar care depaseste cu mult limita adreselor IPv4.

Datorita acestei probleme s-au luat masuri de incetinire a "consumului" de adres IPv4 prin tehnici precum NAT (si totodata introducerea conceptului de IP Public si IP Privat). O alta masura, mult mai bine fata de NAT, este introducerea protocolului IPv6, despre care vom discuta mai tarziu.

Structura Pachetului IPv4

In figura 5.4 poti vedea cum este structura un pachet IPv4:

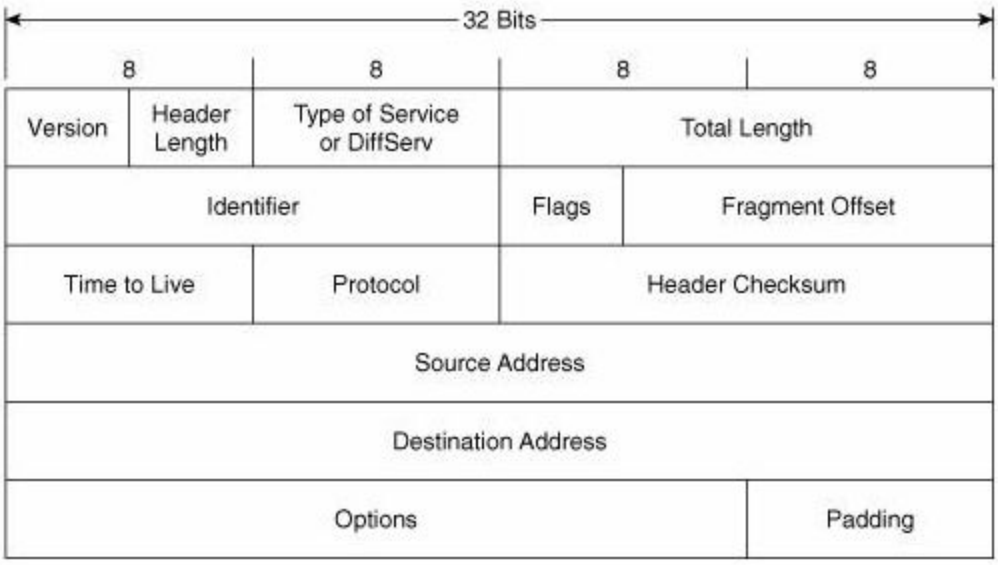


Figura 5.4

Astfel putem identifica cateva elemente importante cu care ne vom confrunta in foarte multe situatii de-a lungul carierei noastre:

- **IP Source Address** (Adresa IP sursa)
- **IP Destination Address** (Adresa IP destinatie)
- **Time to Live** (**TTL**)
- **ToS** (Type of Service)
- **Header Checksum**

Acum hai sa discutam mai in detaliu despre fiecare dintre acestea si vom incepe cu adresele IP. Presupun ca pana aici este clar ca in orice comunicatie dintre 2 dispozitive avem nevoie de o adresa sursa si de o adresa destinatie. In acest caz apar cele 2 campuri (Source & Destination Address) care sunt rezervate pentru **adresa IP sursa** si adresa **IP destinatie**.

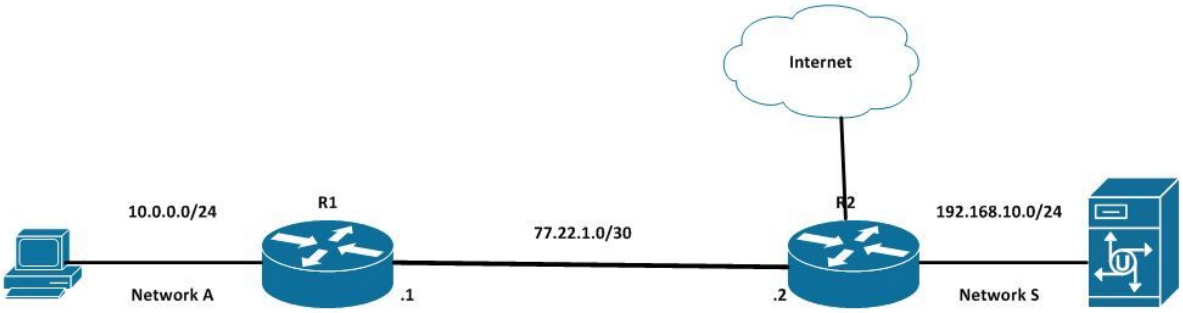


Figura 5.5

De exemplu, in figura 5.5 poti vedea cele 2 retele A si S. Daca PC-ul din reteaua A (cu IP-ul 10.0.0.5) vrea sa comunice cu serverul din reteaua S (cu IP-ul 192.168.10.8), atunci sursa pentru fiecare pachet in parte va fi **10.0.0.5**, iar destinatia **192.168.10.8**.

Clasele de IP-uri

Dupa cum am spus si la inceputul acestui capitol, fiecare camp (4 in total) al unei adrese IP poate avea orice valoarea intre 0 - 255 (8 biti/camp, deci in total 256 de valori; 2^8 = 256). Astfel, adresele IP se imparte in mai multe clase:

Clasa Adresa IP	Adresa IP Start	Adresa IP End	Prefix Retea
A	1.0.0.0	127.255.255.255	1 - 127
B	128.0.0.0	191.255.255.255	128 - 191
C	192.0.0.0	223.255.255.255	192 - 223
D	224.0.0.0	239.255.255.255	224 - 239
E	240.0.0.0	255.255.255.255	240 - 255

Clasele A, B, C sunt **cele folosite** in Internet, clasa D fiind rezervata pentru Adresele de tip Multicast, iar clasa E este o clasa experimentala si nu este folosita.

IP Public vs IP Privat

IP-urile Publice, dupa cum le spune si numele, sunt folosite pentru a comunica (tranzita) in Internet, iar cele Private sunt folosite in Retelele Locale (LAN), cum ar fi reteaua noastra de acasa.

Astfel, IP-urile Private nu vor ajunge niciodata in Internet, deoarece se foloseste un procedeu numit NAT (Network Address Translation) care "transforma" IP-urile Private in IP-uri Publice.

IP-uri Private

Dintre aceste clase se disting urmatoarele **IP-uri PRIVATE**:

Clasa Adresa IP	Adresa IP Start	Adresa IP End	Adresa Retea
A	10.0.0.1	10.255.255.255	10.0.0.0/8
B	172.16.0.1	172.31.255.255	172.16.0.0/12
C	192.168.0.1	192.168.255.255	192.168.0.0/16

NOTA: Restul adreselor IP sunt PUBLICE !

Astfel noi putem avea un scenariu similar cu cel din figura 5.4 de mai jos (mai multe retele locale - LAN - care contin adrese IP private, iar restul retelelor (publice / din Internet) cu adrese IP publice)

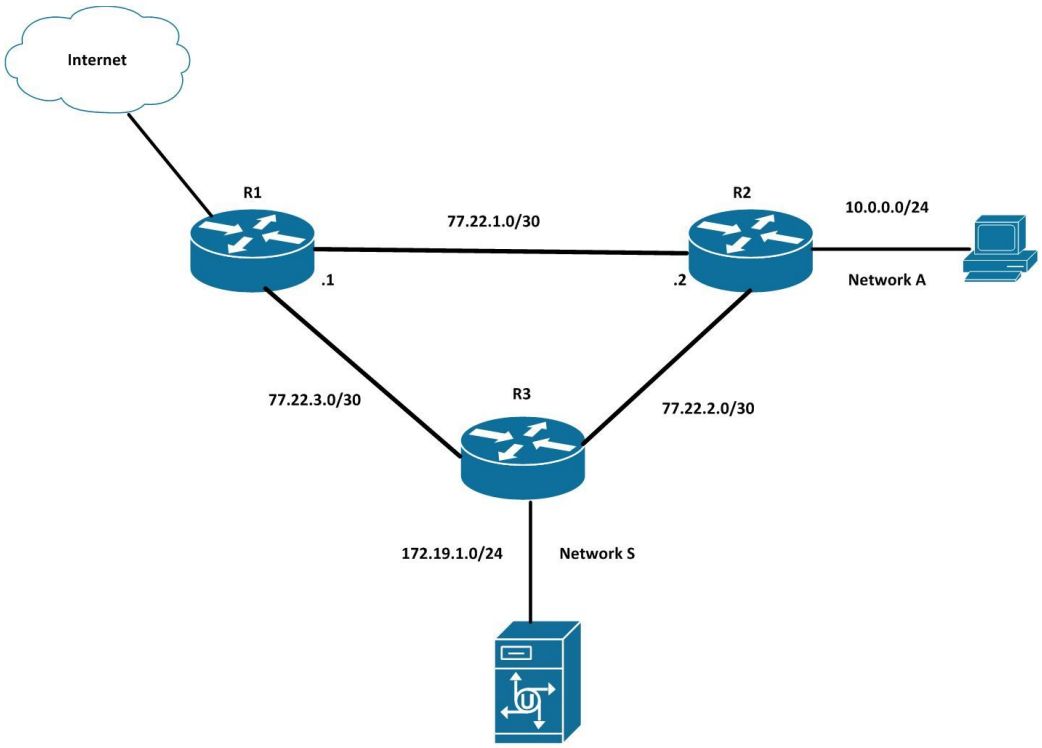


Figura 5.6

Ce este NAT (Network Address Translation) ?

Organizatiile care "administreaza" Internetul au propus, prin conventie, adresele IP Private sa nu poata fi rutate in Internet (**orice pachet cu IP-ul sursa Privat va fi aruncat !**).

Astfel toate companiile furnizoare de servicii de Internet au implementat politici de filtrare a traficului (ACL) pe baza IP-ului sursa care verifica daca un pachet are un IP este privat sau nu. In cazul in care, IP-ul sursa este privat (i.e: 10..., 172.16... sau 192.168...) atunci acesta va fi oprit si "aruncat la gunoi", transportarea lui catre destinatie nefiind permisa.

AICI intervine NAT:

Network Address Translation (**NAT**) mascheaza ("*translateaza*") un IP Privat intr-un IP Public

Practic fara acest mecanism nu am putea accesa internetul. **De NAT se ocupa Routerul** (fie ca este vorba de cel *al companiei tale* sau *a celei la care lucrezi* sau ca este vorba de *Routerul Wireless* din sufrageriile noastre)

Tipurile de NAT

Exista mai multe tipuri de NAT printre care identificam:

1. **NAT Static**
2. **NAT Dinamic**
3. **PAT (Port Address Translation)**

Iar acum, haide sa le luam pe rand si sa vorbim despre fiecare in parte:

1) NAT Static

Face o **mapare 1-la-1** a unui IP Privat intr-un IP Public.

PC2: **192.168.1.5 -> 42.4.51.8**

Este folosit, de obicei, in momentul in care avem un server (Web, FTP, etc.) in reteaua locala (**LAN**) si dorim ca resursele de pe acel server (pagina Web, Serverul de CS, un fisier, etc.) sa fie accesibile din Internet.

Exemplu: Ai creat un folder cu poze din ultima vacanta pe care doresti sa le impartasesti cu prietenii si familia ta. Te-ai gandit sa apelezi la un server web si pentru ca ai o adresa IP Public in plus de la Furnizorul de Internet, ai decis sa apelezi la NAT Static.

Adresa IP a serverului tau este 192.168.1.5, iar cea publica este 93.1.8.6. Faci setarile pe Routerul tau de acasa (din Interfata Web - Browser) si le trimiti prieteniilor si familiei link-ul http://93.1.8.6/poze_vacanta2018, iar acestia vor putea sa iti vada cu succes pozele :)

2) NAT Dinamic

NAT-ul Dinamic face o **mapare m-la-n** a unui IP Privat intr-un IP Public, unde m nu este neaparat egal cu n.

PC1: **192.168.1.6 -> 23.47.5.7**

PC2: **192.168.1.7 -> 23.47.5.8**

PC3: **192.168.1.8 -> 23.47.5.9**

NAT-ul dinamic *foloseste un spatiu de adrese* (ex: 93.1.8.7 pana la 93.1.8.10) pe care le poate aloca cate unui singur calculator care doreste sa ajunga in Internet. Functioneaza pe principiul **FIFO** (primul venit, primul servit), asadar daca avem **20** de PC-uri in retea si

numai **4 adrese IP** publice disponibile, ***doar 4 din cele 20 de PC***-uri vor putea ajunge (comunica) in Internet.

3) PAT (Port Address Translation)

In cazul PAT, **maparea este n-la-1**. Adica, avem mai multe adrese IP Private si te "transformam" intr-o singura adresa IP Publica la care **adaugam Portul Sursa** al conexiunii.

PC1: **192.168.1.6:22413 -> 23.47.5.5:22413**

PC2: **192.168.1.7:62459 -> 23.47.5.5:62459**

PAT ascunde mai multe device-uri (cu IP Privat) in spatele unui singur IP Public.

O conexiune dintre 2 device-uri in Internet contine si urmatoarele elemente:

- *IP Sursa*
- *IP Destinatie*
- *Port Sursa*
- *Port Destinatie*

Cand vine vorba de PAT, Routerul va folosi adresa **IP Sursa** (Privata) si **Portul Sursa**, pentru a identifica conexiunea (exact cum este ilustrat in exemplul de mai sus).

PAT este cea mai folosita varianta de NAT, fiind configurat pe marea majoritate a Routerelor Home-Oriented (TP-Link, D-Link, Asus, Huawei, Cisco etc.).

3 Modalitati de transmitere a Pachetelor prin Retea

Cand vorbim de comunicarea dispozitivelor dintr-o Retea, avem la dispozitie 3 modalitati de trimitere a pachetelor:

- *Unicast*
- *Multicast*
- *Broadcast*

In modul ***Unicast***, comunicarea intre 2 dispozitive este **1 la 1**. Adica exista o sursa si o singura destinatie. Gandeste-te la unicast ca la o convorbire cu un prieten (te adresezi unei singure persoane).

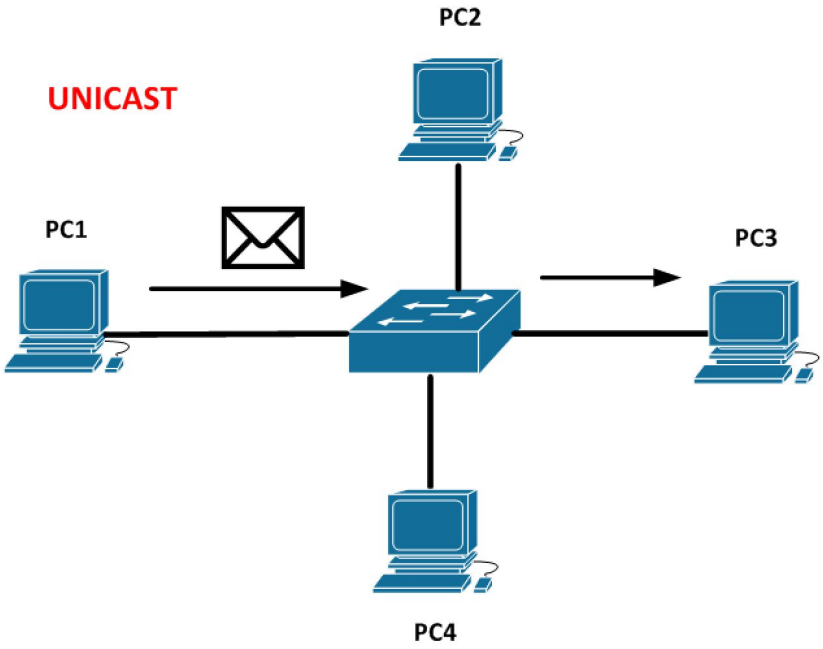


Figura 5.7

In modul ***Multicast***, comunicarea intre dispozitive este **1 la g (grup specific de dispozitive)**. Imagineaza-ti ca te afli intr-o sala cu 100 de persoane, iar tu porti o convorbire doar cu un grup de 10 persoane / colegi (aka. grup specific). Asta inseamna *multicast*.

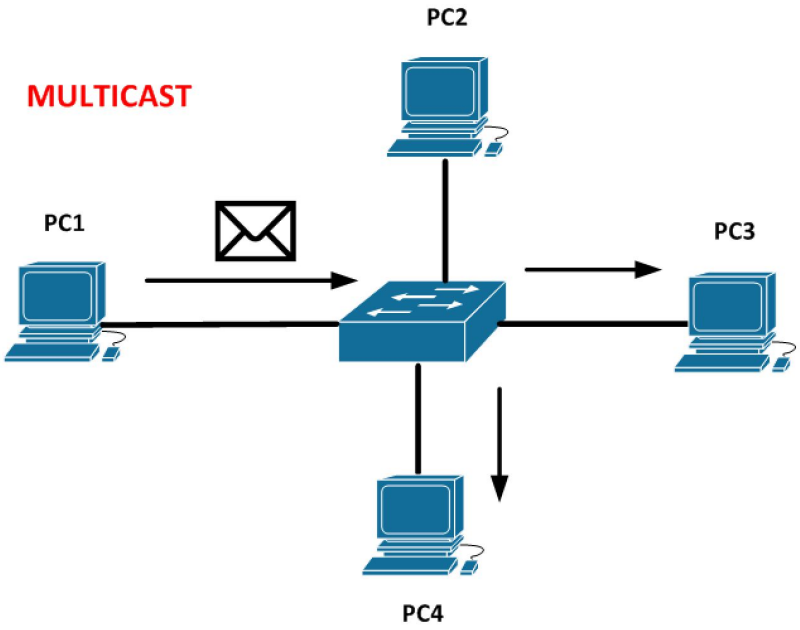


Figura 5.8

In modul ***Broadcast***, comunicarea intre dispozitive este **1 la n** (unde n reprezinta toate dispozitivele din retea). Traficul de tip broadcast este **destinat pentru fiecare dispozitiv** din retea. Inca o data, imagineaza-ti ca te afli in aceeasi sala cu 100 de persoane, iar tu te afli pe scena si le vorbesti tuturor. Acesta este echivalentul broadcastului.

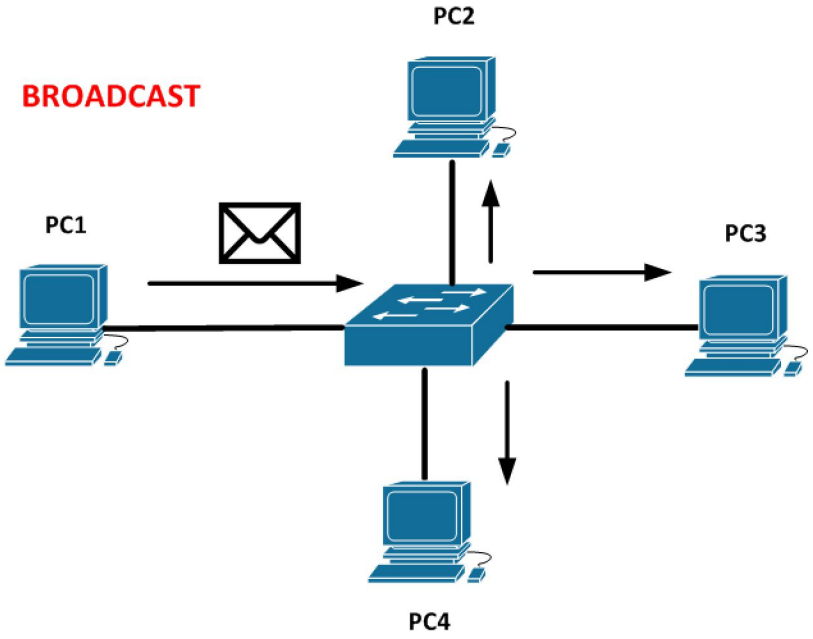


Figura 5.9

2) Structura Pachetului IPv4

In figura 1.2 poti vedea cum este structura un pachet IPv4:

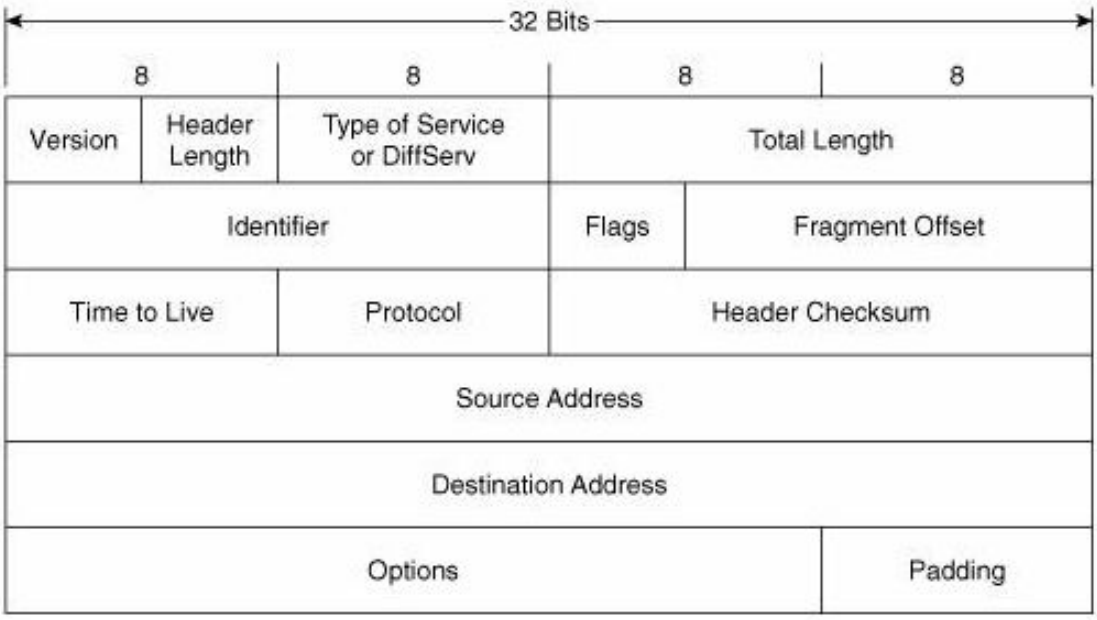


Figura 1.2

Astfel, putem identifica cateva elemente importante cu care ne vom confrunta in foarte multe situatii de-a lungul carierei noastre:

- **IP Source Address** (Adresa IP sursa)
- **IP Destination Address** (Adresa IP destinatie)
- **Time to Live** (**TTL**)
- **ToS** (Type of Service)
- **Header Checksum**

Acum hai sa discutam mai in detaliu despre fiecare dintre acestea si vom incepe cu adresele IP.

Presupun ca pana aici este clar ca in orice comunicatie dintre 2 dispozitive avem nevoie de o adresa sursa si de o adresa destinatie.

In acest caz apar cele 2 campuri (Source & Destination Address) care sunt rezervate pentru **adresa IP sursa** si adresa **IP destinatie**.

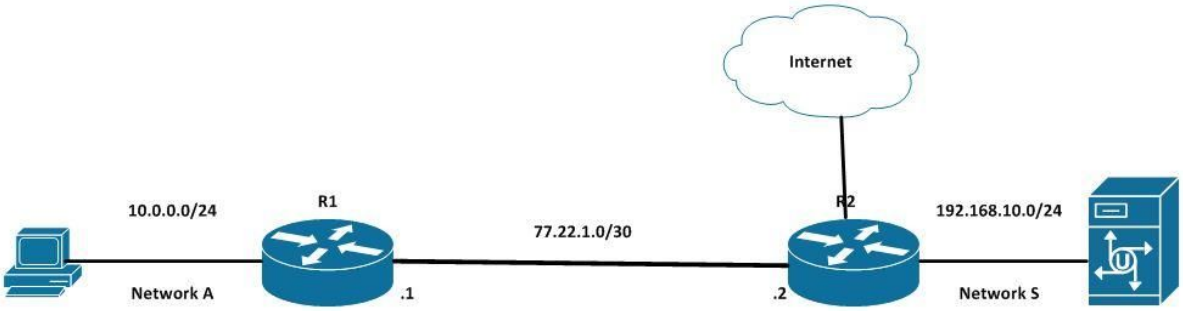


Figura 1.3

De exemplu, in figura 1.3 poti vedea cele 2 retele A si S. Daca PC-ul din reteaua A (cu IP-ul 10.0.0.5) vrea sa comunice cu serverul din reteaua S (cu IP-ul 192.168.10.8), atunci sursa pentru fiecare pachet in parte va fi **10.0.0.5**, iar destinatia **192.168.10.8**.

Oprirea Buclelor dintre Retele cu TTL

Mergand mai departe, avem un alt element foarte important care se numeste **TTL** (Time to Leave/Live). Acesta este un mecanism de protectie impotriva buclelor din Internet care atribuie o anumita valoare fiecarui pachet in parte.

By default, aceasta **valoare** este de **255** per **pachet** (dar poate fi setata intre 1 - 255), iar **acest numar scade** de fiecare data cand ajunge la un alt Router (sau intr-o alta retea). Astfel, daca ar fi sa luam in figura de mai sus, cand

PC-ul trimite un pachet catre serverul S, acesta:

- Ajunge la **R1** cu valoarea **255**
- Ajunge la **R2** cu valoare **254**
- Ajunge la **serverul S** cu valoarea **253**

Daca ar fi avut loc o eroare de configurare pe R2, iar acesta ar fi crezut ca serverul se afla direct conectat la R1, atunci s-ar fi creat o **bucla de retea** pentru ca R2 iar fi trimis pachetul inapoi lui R1, iar R1 l-ar fi trimis inapoi catre R2 (crezand ca serverul este pe partea lui R2). Acest proces ar fi continuat la infinit daca nu ar fi existat TTL-ul.

In momentul in care valoarea **TTL va fi egala cu 0**, **Routerul** care va primi pachetul il va **arunca** (drop). In figura 1.4 poti vedea mai multe valori ale TTL-ului care vin de la mai multe surse din Internet (Yahoo, Google, CoreNetworkz). Dupa cum poti vedea, cel mai usor mod de a afla acest TTL este prin folosirea comenzii **ping.**

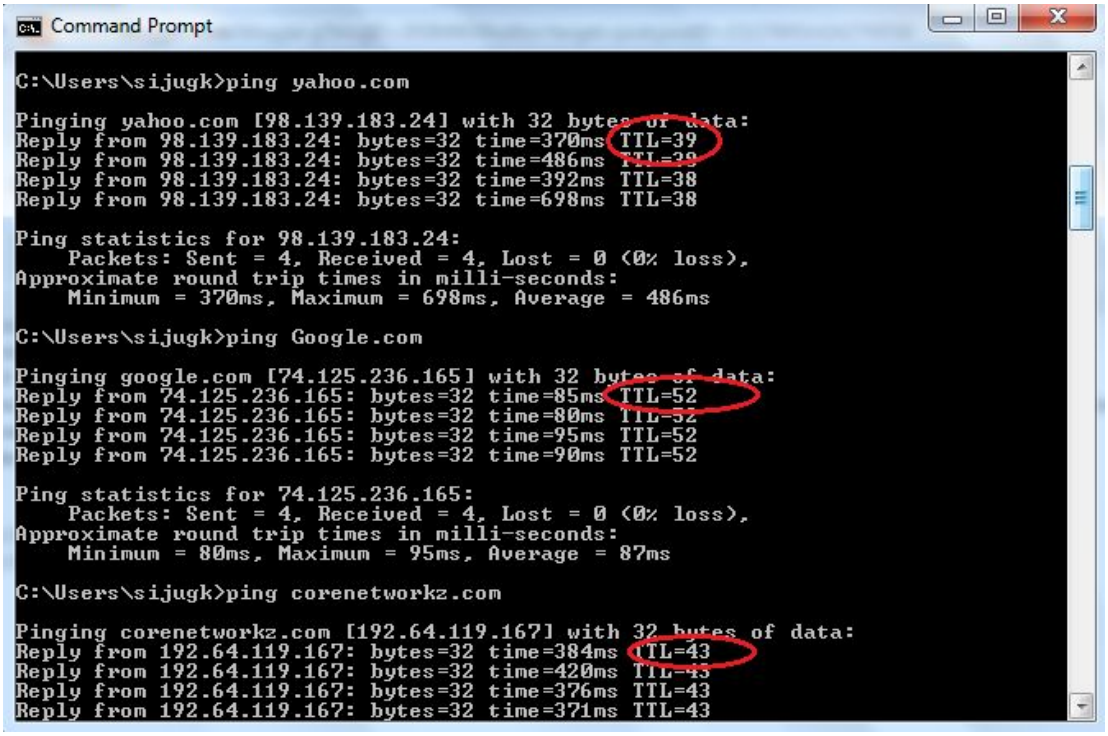


Figura 1.4

Versiunea, ToS si Header Checksum

a) Versiunea

Cand vine vorba de **versiunea** pachetului IP, lucrurile sunt simple pentru ca avem doar 2 variante:

1. IP versiunea 4
2. IP versiunea 6

Noi vorbim in aceast prim capitol doar despre **IPv4**, urmand ca in partea a 2-a a acestei carti sa vorbim si despre **IPv6**.

b) ToS (Type of Service)

ToS este un element important care tine de QoS (Quality of Service). Practic acest camp marcheaza daca un pachet trebuie sau nu sa fie tratat intr-un mod special (acces prioritar/VIP :D). In ce situatii ne ajuta un astfel de feature ? In situatile in care avem **trafic de voce** sau video in retea (ex: convorbiri pe Skype - in general Skype for Business, WebEx). **QoS** (sau ToS) ne garanteaza calitatea serviciilor acest aplicatii **real-time.**

c) Header Checksum

Checksum-ul unui pachet ajuta la **stabilirea integritatii** acestuia. In momentul in care dorim sa trimitem un pachet prin Internet, pe tot parcursul drumului acesta poate suferi modificari (pierderea de pachete, alterarea informatiei, un hacker ii schimba continutul). Astfel, avem nevoie de un mecanism prin care sa ne asiguram ca acesta ramane intact (sau ca **pachetul ajuns** la destinatie este **exact ca cel trimis** de la sursa).

Checksum-ul ne ajuta sa indeplinim acest obiectiv, practic el este o formula matemica care obtine un ID unic pentru orice pachet introdus in aceasta formula. Astfel in momentul in care **Sursa** (PC-ul A) vrea sa trimita pachete catre **Destinatie** (Serverul S) fiecare sunt trecute prin aceasta formula, iar valoarea lor este adaugata in campul “Header Checksum” din IPv4. In momentul in care aceste pachete ajung la destinatie, Destinatia recalculeaza aceste ID-uri pentru fiecare pachet in parte. Daca cele 2 valori (sau ID-uri) - trimise / primite - sunt la fel atunci pachetul este la fel (aka. si-a **pastrat integritatea**).

Alte elemente ale pachetului IP

Iata si restul elementelor care tin de pachetul IP, explicate pe scurt:

- **Total Length** - specifica lungimea totata (in bytes) a pachetului
- **Flags / Fragments Offset** - specifica daca este necesara fragmentarea unui pachet datorita dimensiunii sale ridicate
- **Options** - adaugarea de optiuni aditionale pachetului IP - foarte rar folosit
- **Protocol** - specifica protocolul de la nivelul superior (TCP sau UDP)
- **Identifier** - identificator unic al pachetului IP
- **Padding** - folosit pentru “reglarea” lungimii pachetului astfel incat sa fie multiplu de 32 (bits)

3) Subnetarea retelelor IPv4

Subnetarea este foarte importanta cand vorbim de Design-ul si functionalitatea (la parametri optimi) unei retele. In cazul in care nu este gandit intr-un mod organizat si eficient poate duce la cresterea costurilor si o scadere a eficientei rutarii in retea.
O adresa **IPv4 este alcatuita din 32 de biti**. Astfel fiecare ea va contine 4 campuri fiecare a cate 8 biti. **Fiecare camp** poate avea o valoare intre **0 si 255**, in *total 256 de valori* (pentru ca **2^8 = 256**). Iata un exemplu clasic de adresa IP: ***192.168.0.0/24***

- **192.168.0.0** = **adresa de retea** (gandeste-te la ea ca la *denumirea unei strazi*)
- **/24** = **masca de retea** (te poti gandi la masca, ca fiind *numarul maxim de case* care pot fi *construite* pe o strada).
- **Masca de retea** = indica **dimensiunea** unei retele (numarul total de IP-uri care alcatuiesc reteaua)

Cand vorbim de subnetare/adresare IP o adresa IP este compusa din:

1. **Portiunea de Network** (Retea) - aka. Masca de retea
2. **Portiunea de Host** - aka. diferenta: **32 - Masca**

Iar acum, 192.168.0.0/24 fiind adresa de retea (numele strazii):
/24 - reprezinta masca, adica **primii 24 de biti** reprezinta **portiunea de network** (retea), iar restul de **8 biti** reprezinta **portiunea de host.**

Asadar, daca dorim sa alocam (setam pe dispozitive) adrese IP din reteaua 192.168.0.0, atunci **primele 3 campuri** vor ramane la fel (**192.168.0**), iar **ultimul camp** va identifica fiecare dispozitiv (**de la .1 la .254**):

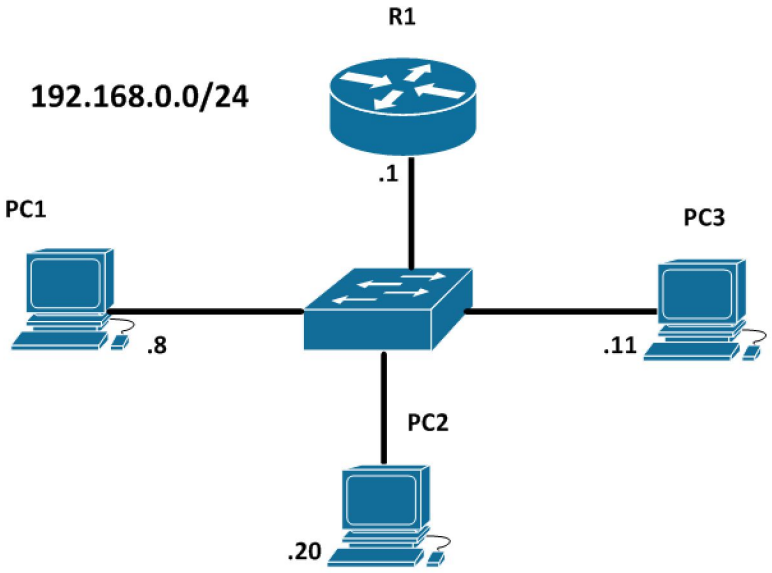


Figura 1.5

Exemplu: 3 dispozitive carora le vom aloca o adresa IP din reteaua 192.168.0.0/24

Primul dispozitiv: *192.168.0.8*

Al 2-lea dispozitiv: *192.168.0.11*

Al 3-lea dispozitiv: *192.168.0.20*

Pe langa asta, de obicei **Routerul** are *prima adresa* (192.168.0.1) sau *ultima adresa* (192.168.0.254) din retea.

Asadar IP-ul lui R1 va fi: *192.168.0.1*

Subnetarea unei Retele

Pentru a putea face subnetarea unei retele, trebuie mai intai sa raspundem la urmatoarele intrebari:

1) **Cate IP-uri** pot avea intre retea cu o masca data ? (ex: /24)

2) Care este **primul** si **ultimul IP** ?

3) **Care sunt adresele IP** utilizabile din retea ?

Exemplu #1

Sa luam reteaua **192.168.0.0/24** si sa raspundem la prima intrebare:

1) 32 - 24 = 8, 2^8 = **256** - numarul maxim de IP-uri dintr-o retea cu masca /24

2) **Primul IP**: 192.168.0.1, **Ultimul IP**: 192.168.0.255 (Bcast)

3) Toate IP-urile in intervalul *192.168.0**.1*** - *192.168.0**.254***, sunt adresele utilizabile din retea.

Asadar, reteaua 192.168.0.0/24 contine urmatoarele adrese IP:

192.168.0.1, 192.168.0.2, 192.168.0.3 ... 192.168.0.255

Dar dintre acestea, *ultima adresa IP* (**192.168.0.255**) nu poate fi folosita pentru ca ea este considera **adresa** de **broadcast.** Astfel vor fi **disponibile doar 254** de adrese IP, din cele 256 pentru ca prima adresa (192.168.0.0 - adresa de retea) si ultima adresa (192.168.0.255 - broadcast) nu pot fi folosite.

Exemplu #2

Vom lua reteaua **10.222.24.0/24** si vom raspundem la cele 3 intrebari:

1) 32 - 24 = 8, 2^8 = 256 - numarul maxim de IP-uri dintr-o retea cu masca /24

2) **Primul IP:** 10.222.24.1, **Ultimul IP**: 10.222.24.255 (Bcast)

3) Toate IP-urile in intervalul *10.222.24.**1*** - *10.222.24.**254***, sunt adresele utilizabile din retea.

Exemplu #3

Vom lua reteaua **172.22.9.0/27** si vom raspundem la cele 3 intrebari:

1) 32 - 27 = 5, 2^5 = 32 - numarul maxim de IP-uri dintr-o retea cu masca /27

2) **Primul IP:** 172.22.9.1, **Ultimul IP**: 172.22.9.31

3) Toate IP-urile in intervalul *172.22.9.1* - *172.22.9.30*, sunt adresele utilizabile din retea.

Exemplu #4

Sa luam reteaua **192.168.11.0/30** si sa raspundem la prima intrebare:

1) 32 - 30 = 2, 2^2 = 4 - numarul maxim de IP-uri dintr-o retea cu masca /30

2) **Primul IP:** 192.168.11.1, **Ultimul IP**: 192.168.11.3

3) Toate IP-urile in intervalul 192.168.11**.1** - 192.168.11**.2**, sunt adresele utilizabile din retea.

Pana aici, totul simplu si usor. Te rog sa faci si tu aceste exercitii pe hartie pentru a intelege mai bine conceptul de subneting (asta e doar inceputul :D).

“Care este urmatoarea retea ?”

La majoritatea cursurilor mele, cand predau subnetare, apare inevitabil intrebarea: “Care este urmatoarea retea ?”, si o liniste profunda se aseaza peste sala. Nimeni nu stie ce sa zica (sau mai exact la ce ma refer) asadar hai sa vedem despre ce e vorba:

Cand vine vorba de subnetare, nu ne intereseaza doar retea actuala ci si care poate fi urmatoarea retea dupa aceasta. Daca avem la dispozitie urmatoarea retea:

192.168.0.0/27

32 - 27 = 5, 2^5 = **32,** in acest scenariu **32** indica numarul total de IP-uri dintr-o retea cu masca /27, dar totodata indica si **delimitare** fiecarei retele in parte (sau incrementul, cum imi place mie sa ii spun).

Deci, conform acestui principiu, urmatoarea retea va fi:

192.168.0.32/27, apoi:

192.168.0.64/27

192.168.0.96/27

192.168.0.128/27

Asadar in incremente de 32 pentru o masca /27.

Iata tabelul de mai jos pentru a vedea si alte exemplu:

Reteaua - IP	Primul IP	Ultimul IP	Urmatoarea Retea
192.168.0.0/24	192.168.0.1	192.168.0.255	192.168.1.0/24
10.0.0.0/27	10.0.0.1	10.0.0.31	10.0.0.32/27
10.10.0.128/26	10.10.0.129	10.10.0.191	10.10.0.192/26

Daca avem o masca **/24** numarul total de IP-uri este **256**. Pentru orice camp dintr-un IP valoarea maxima poate fi ***intre 0 - 255*** (**nu** putem aveam valoarea **256**). Nu putem avea scenariul ~~192.168.0.256~~, dar putem avea 192.168.1.0.

Acest exemplu este similar cu urmatorul:

Sa ne gandim ca avem un ceas si este ora 11:59 (aka 192.168.0.255). Peste 1 minut se poate sa avem ora **11:60** ? NU, bineinteles, dar putem avea ora **12:00** (aka 192.168.1.0). Ei bine, asta se intampla si in cazul subnetarii in aceasta situatie:

Oricand avem o retea care **se termina in** (192.168.0)**.255** si vrem sa mergem la urmatoarea retea, atunci aceasta va avea valoarea (192.168)**.1.0**

Subnetarea unei Retele in functie de Numarul Dispozitivelor

Sa presupunem ca in urma acestui curs de retele de calculatoare am primit o oferta de a face design-ul retelei unei firme de consultanta. Aceasta afacere va avea 3 departamente: **Vanzari, Marketing** si **IT**.

Fiecare dintre aceste departamente va avea anumite dispozitive (Laptop-uri, Imprimante, Servere, etc.) care **necesita** cate **o adresa IP** pentru a comunica intre ele si in Internet. Sa presupunem ca avem nevoie de **15** IP-uri pentru departamentul de **Vanzari**, **7** pentru **Marketing** si **129** pentru **IT**. **Spatiul de adrese disponibil** pe care il avem este urmatorul: **10.23.0.0/16**

Acest spatiu de adrese (daca e sa urmam exemplul de mai sus) va cuprinde urmatoarele:
1) **/16 =>** 32 - 16 = 16, 2^16 = **65536 de IP-uri**
2) Primul IP: **10.23.0.1**, Ultimul IP: **10.23.255.255** (Bcast)

3) Toate IP-urile in intervalul **10.23.0.1 - 10.23.255.254**, sunt adresele utilizabile din retea.

Gandeste-te ca fiecare dintre aceste IP-uri au costul de 1$/luna. Scopul nostru este sa facem subnetarea cat mai eficient. Daca nu facem asta, riscam sa platim 65536$/luna doar pe IP-uri !

Asadar noi incepem subnetarea intotdeauna cu reteaua cea mai mare (ca numar de dispozitive).

1. **129 - IT**
2. **15 - Vanzari**
3. **7 - Marketing**

Acum trebuie sa aflam raspunsul la urmatoarele intrebari:

1) Care este **masca** ? - ex: /24
2) Care este **reteaua** ?
3) Care sunt **adresele IP utilizabile** din retea ?
4) Care este **urmatoare retea** ?
Pentru a afla masca de retea trebuie mai intai sa ne punem intrebare "**care este cea mai apropiata putere a lui 2, mai mare decat numarul de dispozitive** (ex: 129) **?"**

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

In cazul *IT-ului*, de **129** cea mai apropiata putere este **2^8** = **256.** Dupa cum spuneam si mai drevreme, **256 reprezinta numarul Maxim de IP-uri**, dar nu le putem folosi pe toate. *Motivul fiind*: prima adresa (**.0**) este **rezervata pentru retea**, iar ultima adresa (.**255**) este **rezervata pentru Broadcast**. Asadar trebuie **sa scadem 2 IP-uri** din totalul de 256, rezultatul fiind **254.**

Din acest numar, ne intereseaza puterea lui 2, in cazul acesta 8. Astfel putem raspunde la intrebarile de mai sus:

1) **32 - 8** = **24** => **/24** reprezinta masca de retea

2) Reteaua va fi chiar **10.23.0.0/24** (dar dupa cum poti vedea, **schimbam masca**).

3) Toate IP-urile in intervalul **10.23.0.1 - 10.23.0.254**, apartin retelei

4) **Urmatoarea** retea: **10.23.1.0/24** (din care putem subneta mai departe)

Asadar, **prima** retea este: **10.23.0.0/24**

In cazul departamentului de *Vanzari*, de **15** cea mai apropiata putere este **2^5 = 32** (mai exact 32 - 2 = **30**). Din acest numar, **ne intereseaza puterea** lui 2, in cazul acesta **5**. Astfel putem raspunde la intrebarile de mai sus:

1) **32 - 5 = 27 => /27** reprezinta masca de retea

2) Reteaua este **10.23.1.0/27**

3) Toate IP-urile in intervalul **10.23.1.1 - 10.23.1.30**, apartin retelei

4) **Urmatoarea** retea: **10.23.1.32/27** (din care putem subneta mai departe - retelele cu masti mai mici)

A **2-a** retea este: **10.23.1.0/27**

In cazul departamentului de *Marketing*, de **7** cea mai apropiata putere este **2^4 = 16** (mai exact 16 - 2 = **14**). Din acest numar, **ne intereseaza puterea** lui 2, in cazul acesta **4**. Astfel putem raspunde la intrebarile de mai sus:

1) **32 - 4 = 28 => /28** reprezinta masca de retea

2) Reteaua este **10.23.1.32/28**

3) Toate IP-urile in intervalul **10.23.1.33 - 10.23.1.46**, apartin retelei

4) **Urmatoarea** retea: **10.23.1.48/28** (din care putem subneta mai departe - retelele cu masti mai mici)

A **3-a** retea este: **10.23.1.32/28**

Iata un rezumat a celor discutate mai sus:

Network	Prima Adresa	Ultima Adresa	Adresa de Bcast	Urmatoarea Retea	Numarul de IP-uri din retea
10.23.0.0/24	10.23.0.1	10.23.0.254	10.23.0.255	10.23.1.0/24	254
10.23.1.0/27	10.23.1.1	10.23.1.30	10.23.1.31	10.23.1.32/27	30
10.23.1.32/28	10.23.1.33	10.23.1.46	10.23.1.47	10.23.1.48/28	14

4) Setarea unei adrese IP in Windows 7/8/10

Pe Windows, cand vine vorba de setare a unei adrese IP, lucrurile stau putin diferit (in sensul ca, adresa IP o vom configura prin GUI). Dar mai intai hai sa verificam adresa IP din CMD:

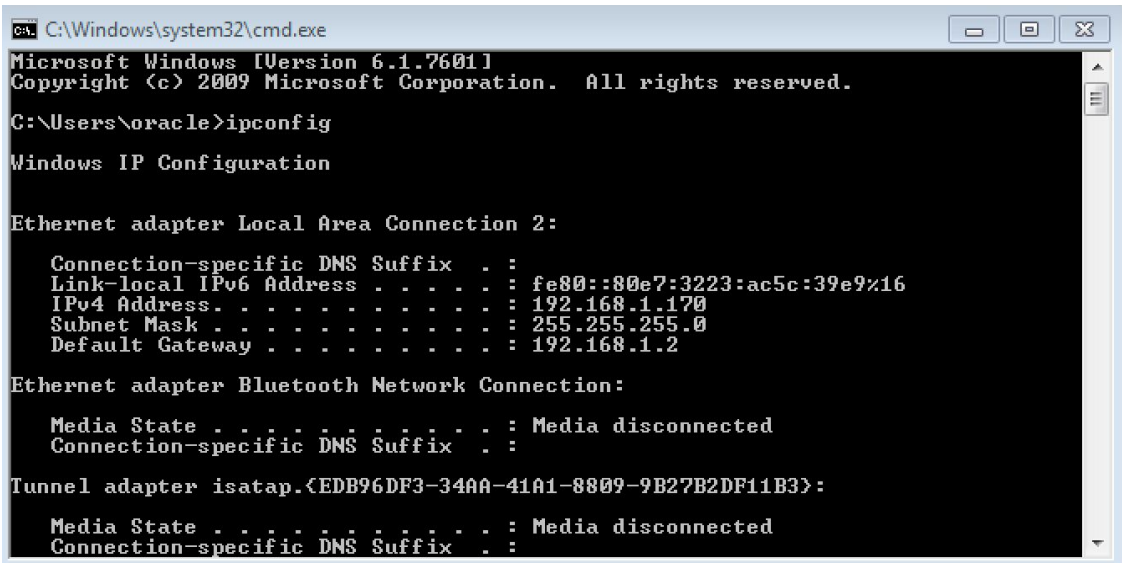


Figura 1.6

Comanda pe care am folosit-o este **>ipconfig** si dupa cum poti vedea ne afiseaza mai multe informatii (*adresa IP, masca de retea, default gateway*, etc) despre interfata Ethernet (LAN), Wi-Fi, Bluetooth si altele (depinde de configuratia OS-ului).

Ok, si pana la urma cum am configura aceasta adresa IP ? Iata in figurile de mai jos cum:

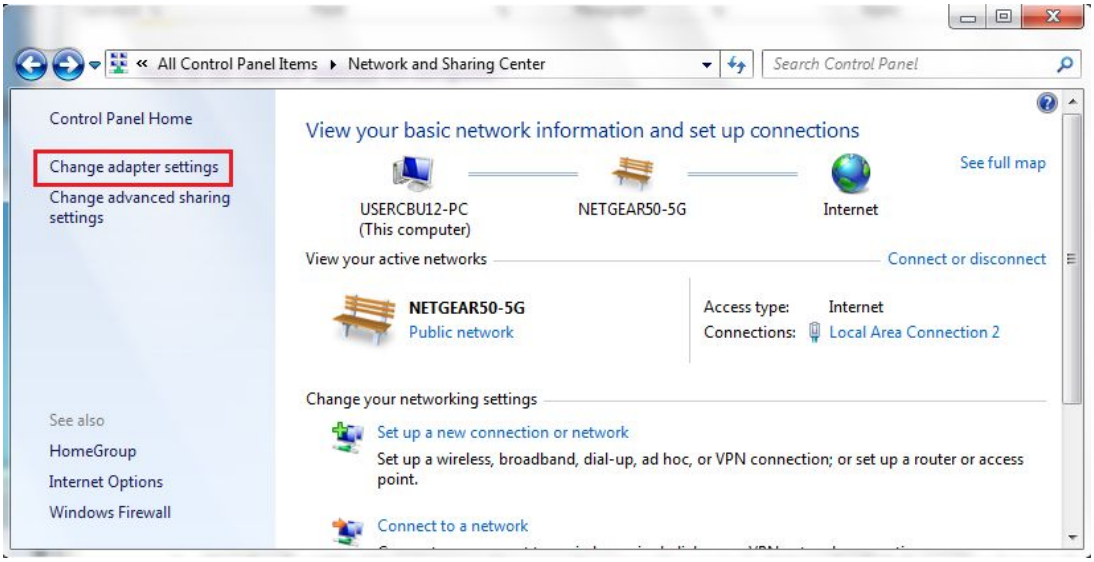


Figura 1.7

Un mod foarte simplu de a face aceasta setare statica a adresei IP este de a merge, mai intai, in **Control Panel -> Network and Sharing Center** dupa care, in partea stanga, "**Change adapter settings**" (sau, alt mod **Network and Internet -> Network Connections**) si vei ajunge la o fereastra similara cu cea care apare in figura de mai jos.

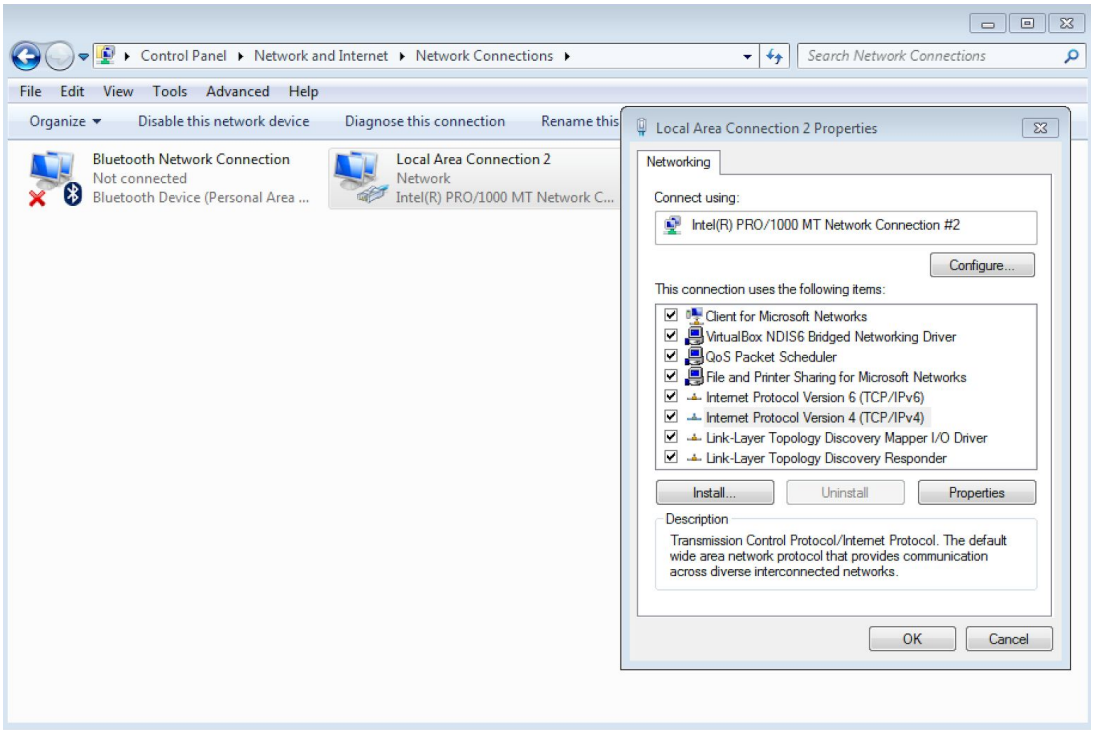


Figura 1.8

Aici ne intereseaza "**Local Area Connection 2**" (in cazul tau poate avea alt nume), pe

care il vom selecta si vom da click dreapta -> Properties. Astfel vom ajunge sa ni se deschida fereastra din partea dreapta unde vom selecta **IPv4** si apoi vom **apasa** pe **Properties**. Iar acum am ajuns la figura de mai jos, lucul in care putem seta (in sfarsit) adresa IP:

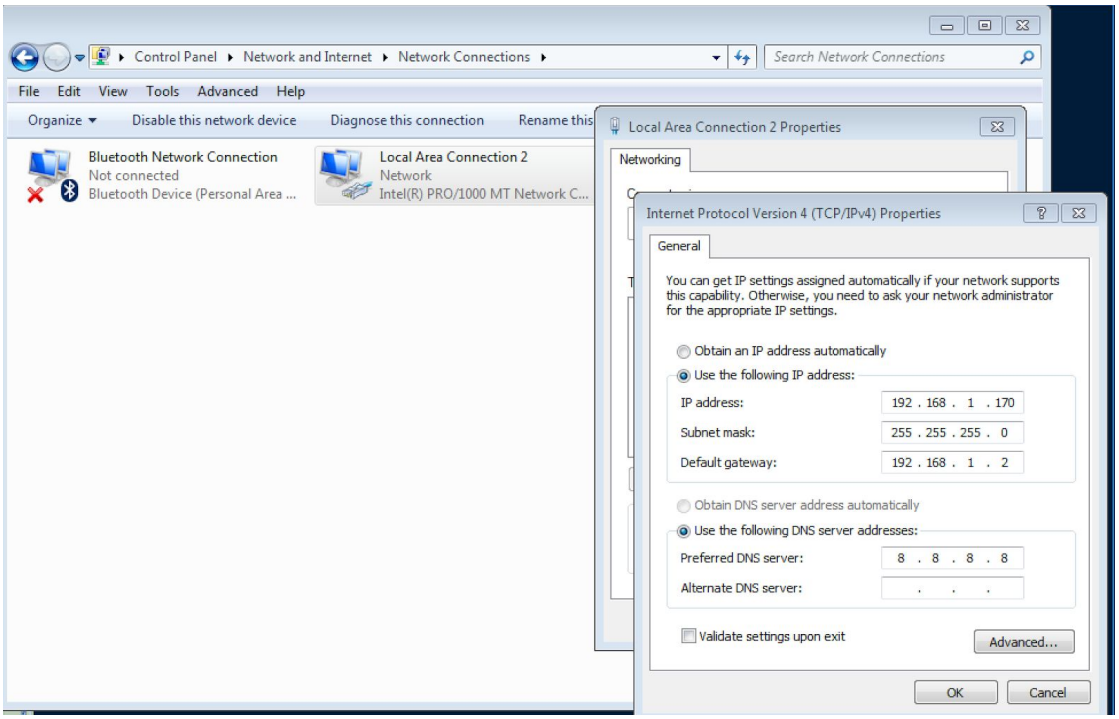


Figura 1.9

Am ales reteaua 192.168.1.0/24, din care **IP-ul 192.168.1.170** l-am asignat PC-ului (Windows 7), **masca** /24 in decimal arata astfel 255.255.255.0, iar **default gateway**-ul (Routerul conectat la Internet) are IP-ul **192.168.1.2**.

De asemenea, am setat si serverul **DNS** (cel care ne ajuta cu rezolvarea de nume: dintr-un domeniu (ex: google.ro) ne va oferi adresa IP a acestuia (ex: 216.58.214.227)) cu IP-ul **8.8.8.8**.

Acum ca am facut toate aceste setari, putem verifica (din CMD) folosind comenzile:

```
>ping 8.8.8.8
```
//verifica conexiunea la Internet (mai exact la 8.8.8.8, care se afla in Internet

```
>ping google.ro
```
//verifica serviciul DNS **si** conexiunea la Internet

```
>nslookup google.ro
```
//verifica serviciul DNS

Exercitiul #1 de Subnetare

Acum a venit momentul sa pui in practica cele discutate mai sus. Te invit sa completezi tabelul de mai jos (pe care il gasesti in materialele de laborator primite).

Network	Prima Adresa	Ultima Adresa	Adresa de Bcast	Urmatoarea Retea
173.22.50.0/24				
13.212.2.128/25				
13.22.1.0/27				
13.22.1.192/29				
92.68.2.64/26				
92.68.2.224/30				
84.88.32.24/29				
84.88.32.34/29				
44.55.66.77/26				
66.55.44.33/28				
11.22.33.44/30				
22.22.22.22/32				
44.11.22.59/25				

Exercitiul #2 de Subnetare

In al 2-lea exercitiu vei face subnetarea unei retele in functie de adresele IP necesare. Nu uita sa ordonezi retelele (cu IP-urile necesare) in ordine descrescatoare (astfel vei incepe de la reteaua cu cele mai multe adrese IP necesare).

Network	Reteaua Din care face Parte	Prima Adresa	Ultima Adresa	Adresa de Bcast	Urmatoarea Retea
Folosind Spatiul de adrese 172.22.96.0/19 alocati intr-un mod cat mai eficient adresele IP pentru urmatoarele retele					
234 IP-uri					
32 IP-uri					
54 IP-uri					
278 IP-uri					
124 IP-uri					
2 IP-uri					
3 IP-uri					
10 IP-uri					
71 IP-uri					
150 IP-uri					
24 IP-uri					
39 IP-uri					
155 IP-uri					

Ce este IPv6 ?

Astfel a aparut nevoia pentru **IPv6**, un nou protocol de adresare care introduce un nou format de adresare (in hexadecimal) si un spatiu mult, mult mai mare de adrese. IPv6 este pe 128 de biti (asta insemnand ca putem avea 2^128 de adrese disponibile) care reprezinta un spatiu infinit mai mare fata de IPv4 care este pe 32 de biti.

Conform Wikipedia, acest numar arata cam asa:

*"**340,282,366,920,938,463,463,374,607,431,768,211,456** sau **3.4×10^{38}** (340 trillioane trillioane trillioane)"*

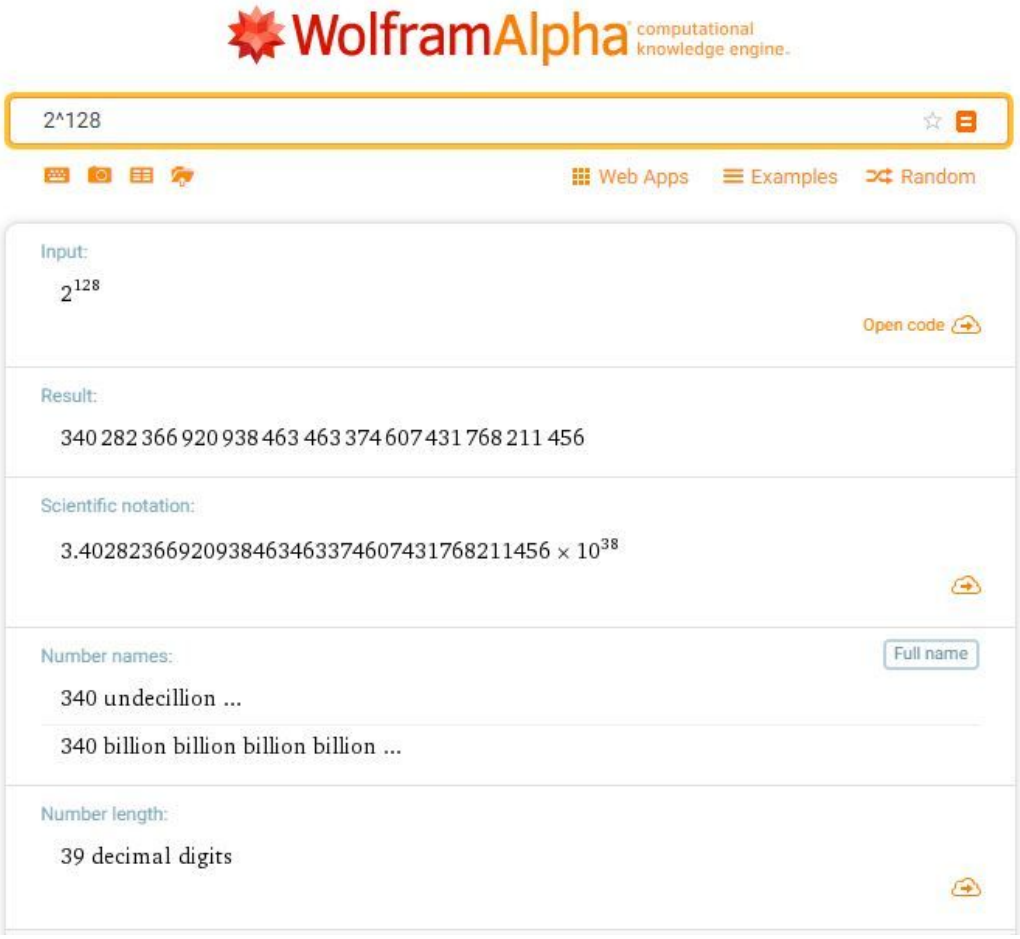


Figura 5.14

Iata cateva exemple de adrese IPv6:

- **2001:DB8:85A3:8D45:119:8C2A:370:734B /64**
- **FE80::C001:37FF:FE6C:0/64**
- **2001::1/128**

Dupa cum poti vedea adresele IPv6 sunt in format **hexadecimal** (include pe langa cifre de la **0 – 9** si literele **A – F**). O adresa IPv6 este compusa din maxim 8 campuri si o masca de retea (care indica cat de mare dorim sa fie reteaua – ca numar de adrese).

Notatii: Fiecare camp al adresei IPv6 este separat de "**:**", dar exista si exceptii:

2002:ABCD:1234:BBBA:0000:0000:0000:0001/64 poate fi scris in mai multe moduri:

a) 2002:ABCD:1234:BBBA**:0:0:0:**1/64

b) 2002:ABCD:1234:BBBA**::**1/64

Daca dorim sa reducem un lant intreg de 0-uri il vom simplifica prin "**::**".

ATENTIE ! "**::**" poate fi folosit **o singura data**.

In figura 5.15, de mai jos, poti sa vezi o adresa IPv6 (prin comanda >ipconfig) din CMD care incepe cu notatia **FE80**:... Aceasta adresa IPv6 este una speciala in sensul ca poate fi folosita doar in reteaua locala (LAN) pentru comunicarea cu celelalte dispozitive.

Aceasta adresa se numeste **Link-Local** si este **generata automat** (alt feature important al IPv6 - autoconfigurarea adreselor).

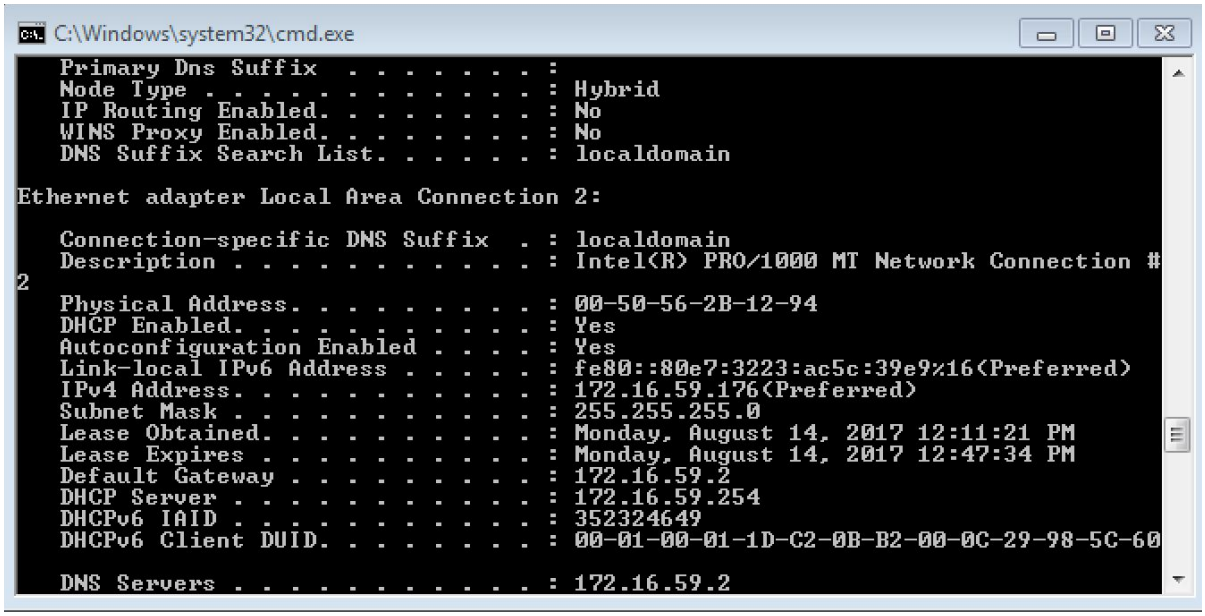


Figura 5.1

In capitolele viitoare vom discuta mult mai in detaliu despre IPv6 si despre cum il putem configura pe echipamente (Routere si PC-uri/Laptop-uri).

Capitolul 6 - Nivelul 4 - Transport

TCP, UDP si Porturi

1) TCP (Transmission Control Protocol)

TCP vine de la **T**ransmission **C**ontrol **P**rotocol si fix asta face, asigura controlul transmisiunilor. El este un protocol care face parte (impreuna cu UDP) din nivelul 4 din modelul OSI, mai exact nivelul Transport. Daca nu esti familiar cu modelul OSI, te invit sa urmaresti tutorialul de mai jos:

TCP-ul este un protocol pe care il folosim tot timpul (fara sa ne dam seama). Spre exemplu in momentul in care descarcam un fisier de pe Internet, sau accesam o pagina web sau **ne conectam in vreun fel la un dispozitiv de retea**, in fiecare dintre aceste situatii folosim acest protocol.

Si acum vine intrebarea: ***De ce ? De ce avem nevoie el ?*** Pentru ca TCP-ul ne permite sa avem conexiunile (sau mai exact datele transmise de la server) exact asa cum ar trebui ele sa fie.

Astfel in momentul in care descarcam un fisier (prin FTP), TCP-ul se va asigura ca fiecare bit din fisierul primit este identic cu fisierul de pe acel server, se va asigura ca pachetele isi pastreaza ordinea, iar in cazul in care se pierd pachete pe drum, le va asigura retransmiterea.

Asadar, acestea sunt o parte din caracteristicile si avantajele protocolului TCP:

- **Asigura retransmiterea** pachetelor (in cazul in care acestea se "pierd pe drum")
- **Ordoneaza pachetele** ajunse la destinatie (acestea pot ajunge diferit fata de cum au fost trimise)
- **Stabileste o conexiune** intre client si server

TCP-ul reuseste sa faca toate acestea cu ajutorul unor tipuri de mesaje speciale:

- SYN, ACK, FIN
- PSH, RST, URG

In figura de mai jos, poti vedea arata header-ul acestui protocol:

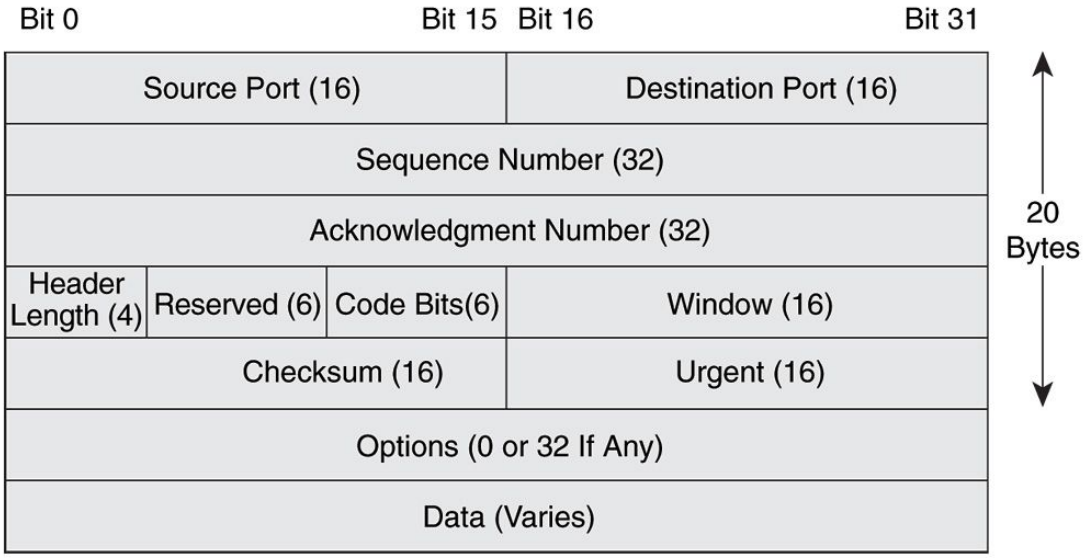


Figura 6.1

Iar in figura urmatoare, poti vedea **cum arata ele in Wireshark**:

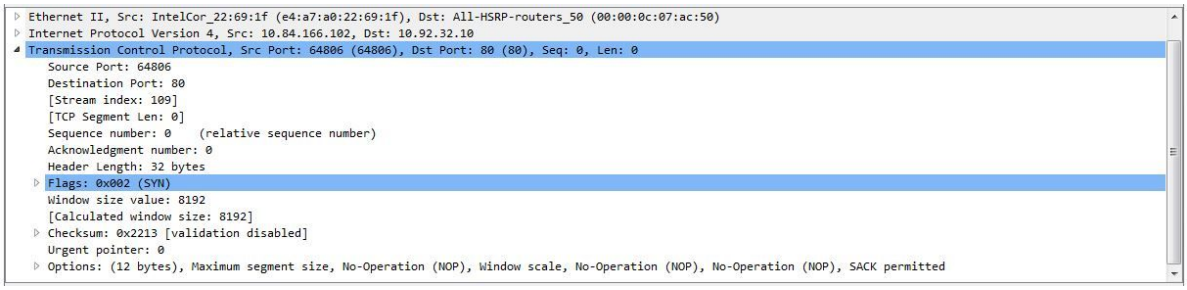


Figura 6.2

Avand toate aceste campuri in header-ul protocolului, TCP-ul ne poate asigura:
TCP *asigura ordinea si retransmiterea* in caz de pierdere de pachete, cu ajutorul **numerelor de secventa (sequence number)**.

Fiecare pachet (sau grup de pachete) are asociate un numar de secventa. Daca destinatarul primeste un anumit numar de pachete (definit de numarul de secventa), atunci acesta va trimite inapoi un mesaj de confirmare (ACK) pentru aceste pachete primite:

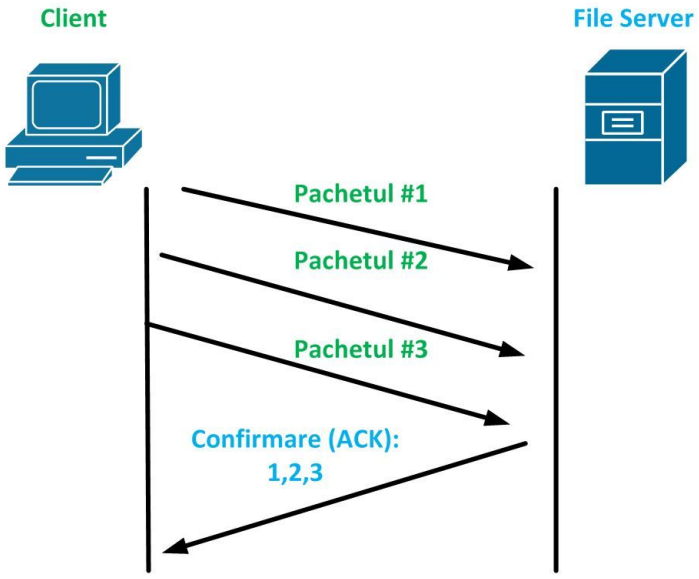


Figura 6.3

Astfel, este usor pentru destinatar sa-si dea seama ce pachete au ajuns la el si ce pachete trebuiesc retransmise. In cazul in care sursa (clientul) nu primeste o confirmare (ACK) pentru vreun pachet, atunci el va retransmite acele pachete.

In momentul in care 2 dispozitive doresc sa trimita trafic intre ele, mai intai, trebuie sa aiba loc o conexiune intre client si server, denumita the **3-Way Handshake**.

Cum stabilim o conexiune intre Client si Server ? (3-Way Handshake)

Dupa cum spuneam si mai devreme, in momentul in care un server trebuie sa comunice cu un client, cei 2 vor forma o conexiune intre ei cu ajutorul 3-Way Handshake-ului, care functioneaza in felul urmator:

In prima faza, Clientul (cel care incepe conexiunea) ii va trimite serverului:

1. un mesaj de sincronizare (**SYN**) sau de incepere a conexiunii
2. serverul va raspunde cu o confirmare (**SYN-ACK**)
3. clientul va raspunde si el cu un mesaj de confirmare (**ACK**)

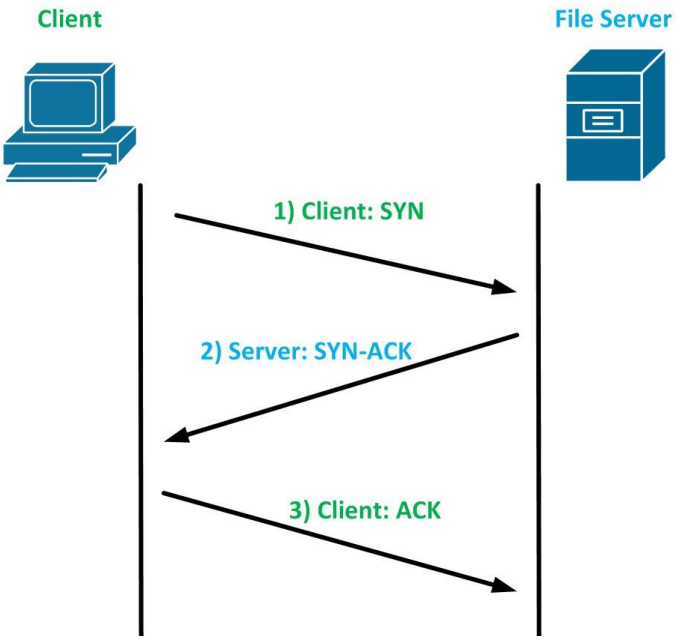


Figura 6.4

Iata o **captura in Wireshark** cu cele 3 pachete prezentate mai sus:

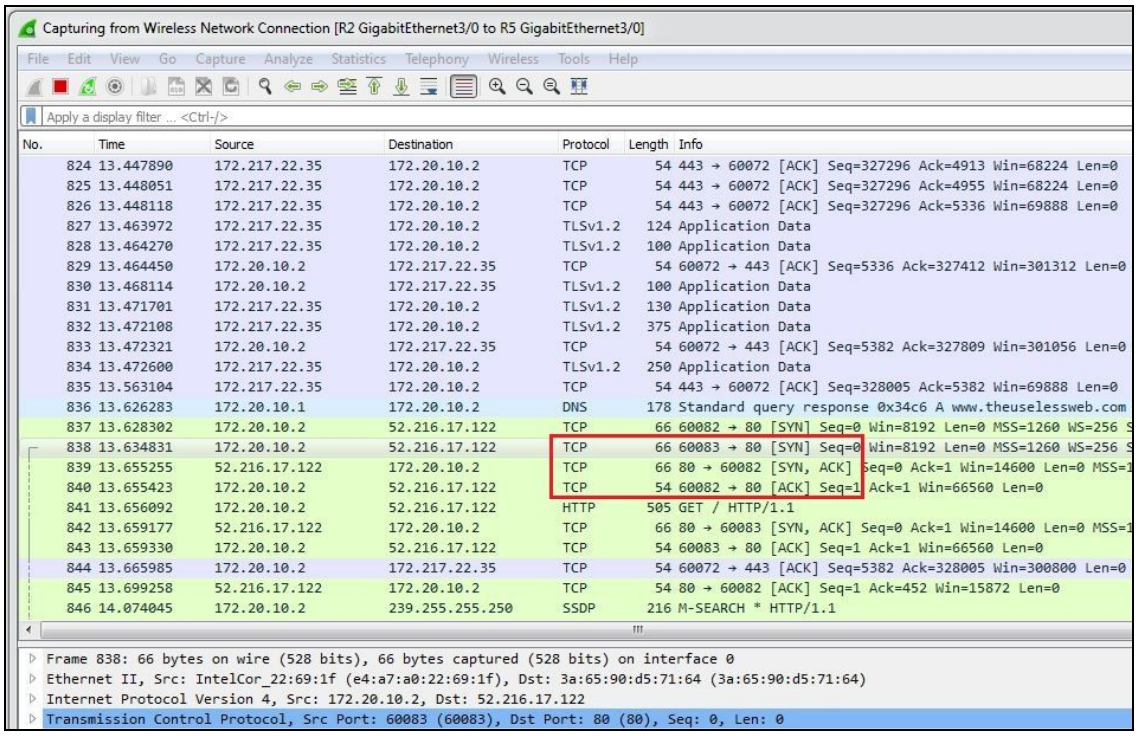


Figura 6.5

Astfel a fost stabilita conexiunea TCP (prin 3-way handshake) intre client si server. Acum cele 2 dispozitive pot comunica (trimite trafic web, transfer de fisier etc.). Acest mecanism ne ajuta sa stabilim o ordine pentru fiecare pachet in parte, sa ne asiguram ca au ajuns toate la destinatie, iar in caz de esec sa retransmitem pachetele.

Cum se incheie conexiunea TCP formata ?

Dupa au fost transmise toate pachetele, conexiunea trebuie sa se incheie. Acest fapt se intampla similar cu cel de la 3-way handshake, doar ca de data aceasta se vor folosi 4 pachete:

1. **clientul** va trimite un **FIN** (mesaj de finalizare/incheiere a conexiunii)
2. **serverul** va raspunde cu un mesaj de confirmare (**FIN-ACK**)
3. Serverul va trimite si el un mesaj **FIN**
4. Iar clientul va raspunde cu un **FIN-ACK**

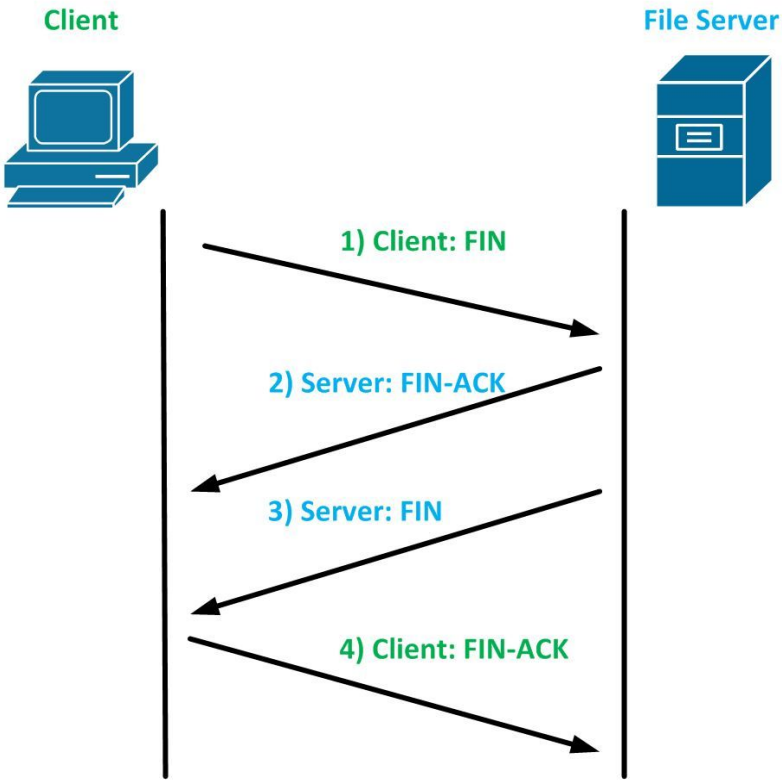
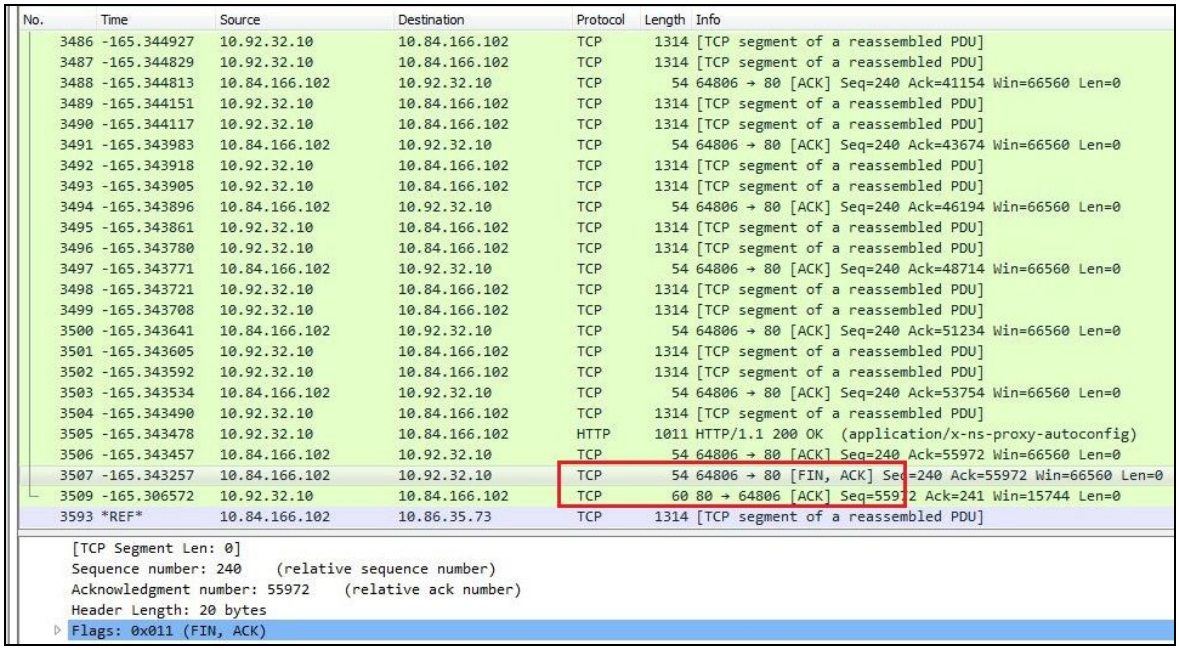


Figura 6.6

lata si **captura in Wireshark** cu conceptele discutate mai sus:



No.	Time	Source	Destination	Protocol	Length	Info
3486	-165.344927	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3487	-165.344829	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3488	-165.344813	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=41154 Win=66560 Len=0
3489	-165.344151	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3490	-165.344117	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3491	-165.343983	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=43674 Win=66560 Len=0
3492	-165.343918	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3493	-165.343905	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3494	-165.343896	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=46194 Win=66560 Len=0
3495	-165.343861	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3496	-165.343780	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3497	-165.343771	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=48714 Win=66560 Len=0
3498	-165.343721	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3499	-165.343708	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3500	-165.343641	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=51234 Win=66560 Len=0
3501	-165.343605	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3502	-165.343592	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3503	-165.343534	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=53754 Win=66560 Len=0
3504	-165.343490	10.92.32.10	10.84.166.102	TCP	1314	[TCP segment of a reassembled PDU]
3505	-165.343478	10.92.32.10	10.84.166.102	HTTP	1011	HTTP/1.1 200 OK (application/x-ns-proxy-autoconfig)
3506	-165.343457	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [ACK] Seq=240 Ack=55972 Win=66560 Len=0
3507	-165.343257	10.84.166.102	10.92.32.10	TCP	54	64806 → 80 [FIN, ACK] Seq=240 Ack=55972 Win=66560 Len=0
3509	-165.306572	10.92.32.10	10.84.166.102	TCP	60	80 → 64806 [ACK] Seq=55972 Ack=241 Win=15744 Len=0
3593	*REF*	10.84.166.102	10.86.35.73	TCP	1314	[TCP segment of a reassembled PDU]

```
[TCP Segment Len: 0]
Sequence number: 240    (relative sequence number)
Acknowledgment number: 55972    (relative ack number)
Header Length: 20 bytes
Flags: 0x011 (FIN, ACK)
```

Figura 6.7

Si astfel conexiunea TCP dintre cele 2 device-uri se va incheia.

2) UDP (User Datagram Protocol)

UDP este *fix opusul TCP-ului* (nu retransmite pachete, nu are un mod de stabilire a conexiunilor, etc.). UDP pur si simplu trimite pachetele de la o anumita sursa catre o destinatie fara sa-l intereseze starea acestora. **Avantajul** folosirii acestui protocol este reprezentat de **latenta scazuta** (delay) si permite fluiditatea aplicatiei fara intarzieri.

Asadar **UDP** este un protocol **potrivit pentru aplicatiile real-time** (ex: Voce, Video) care pur si simplu au nevoie sa ajunga la destinatie cat mai repede posibil. In figura 6.8 poti vedea cum arata headerul UDP-ului, iar in comparatie cu TCP, acesta e mult mult mai simplu si eficient.

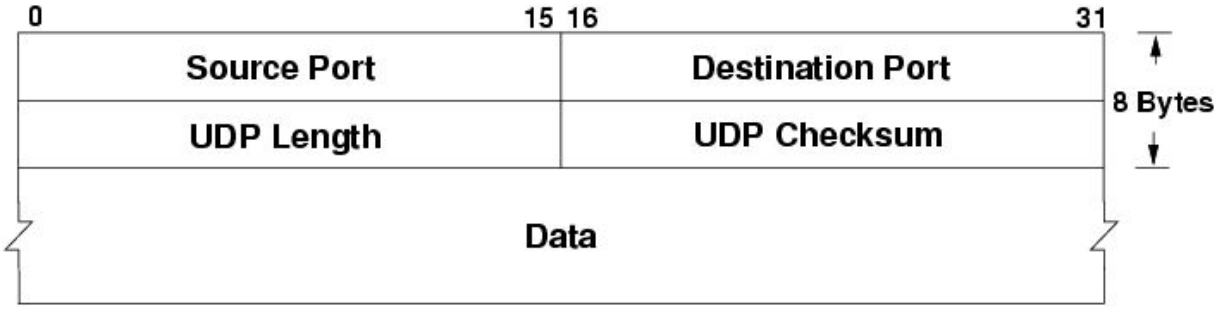


Figura 6.8

Pentru ca tot vorbeam de aplicatii in timp real (real-time), cum ar fi Skype, Facebook Live, CS Online Multiplayer, iata cateva cerinte de functionare la o calitate decenta a unei convorbiri audio prin **VoIP** (**V**oice **o**ver **IP**):

- **Delay: < 150 ms**
  - Deschide CMD si scrie ping 8.8.8.8 pentru a vedea ce delay ai)
- **Pierdere de pachete: < 1%**
  - 1 secunda de voce = 50 pkt de 20 ms audio fiecare => 1% din 50 = 0,5; adica la 2 secunde de audio se poate pierde maxim un pachet)
- **Jitter (delay variabil): < 30ms**

Iata si cateva capturi in **Wireshark** pentru traficul UDP:

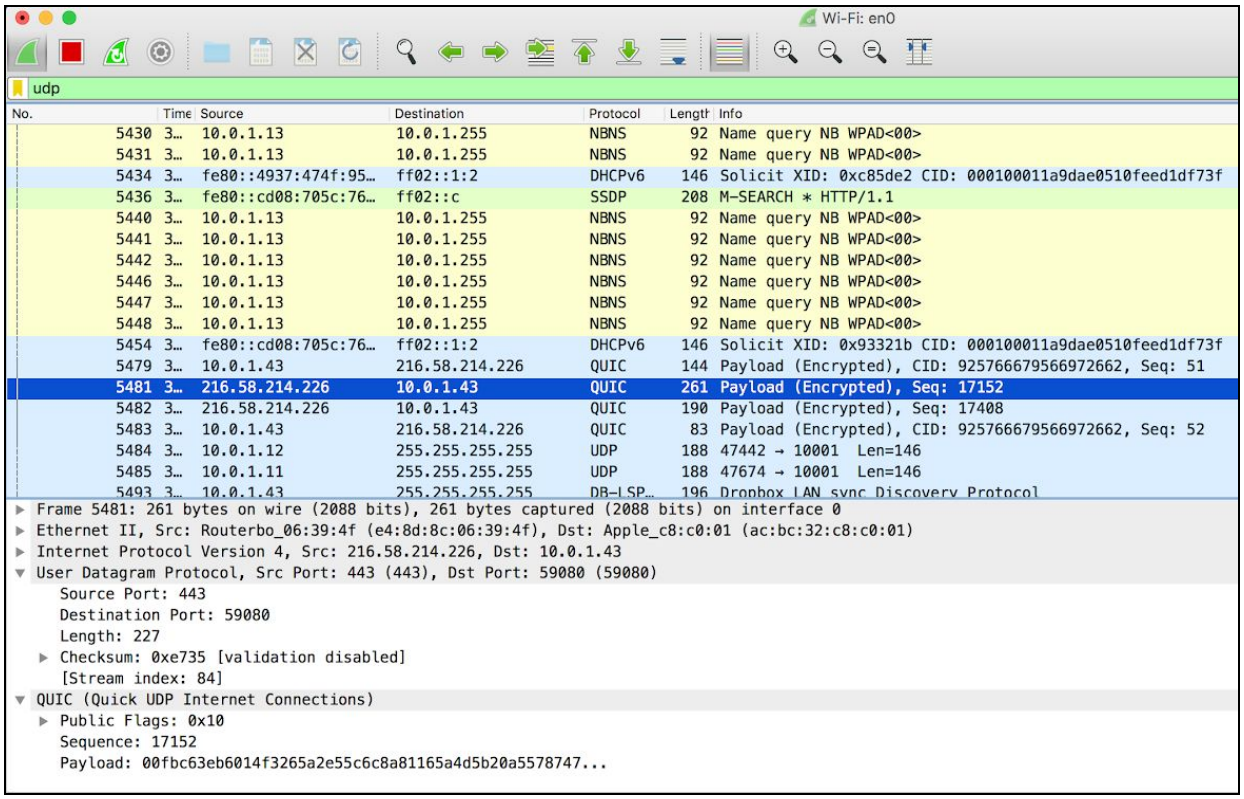


Figura 6.9

In figura 6.9 putem vedea protocolul UDP in actiune. De data aceasta am selectat un protocol numit QUIC (Quick UDP Internet Connections) care activeaza peste UDP (asa cum putem vedea in figura).

El ne ajuta cu **transmiterea traficului** intr-un **mod criptat** (dupa cum poti vedea: *Payload (Encrypted)*).

Inca un lucru pe care te invit sa-l observi este complexitatea redusa a header-ului UDP (pe care am vazut-o si in figura 6.8), mai exact **portul sursa**, **portul destinatie** si lungimea totala a headerului.

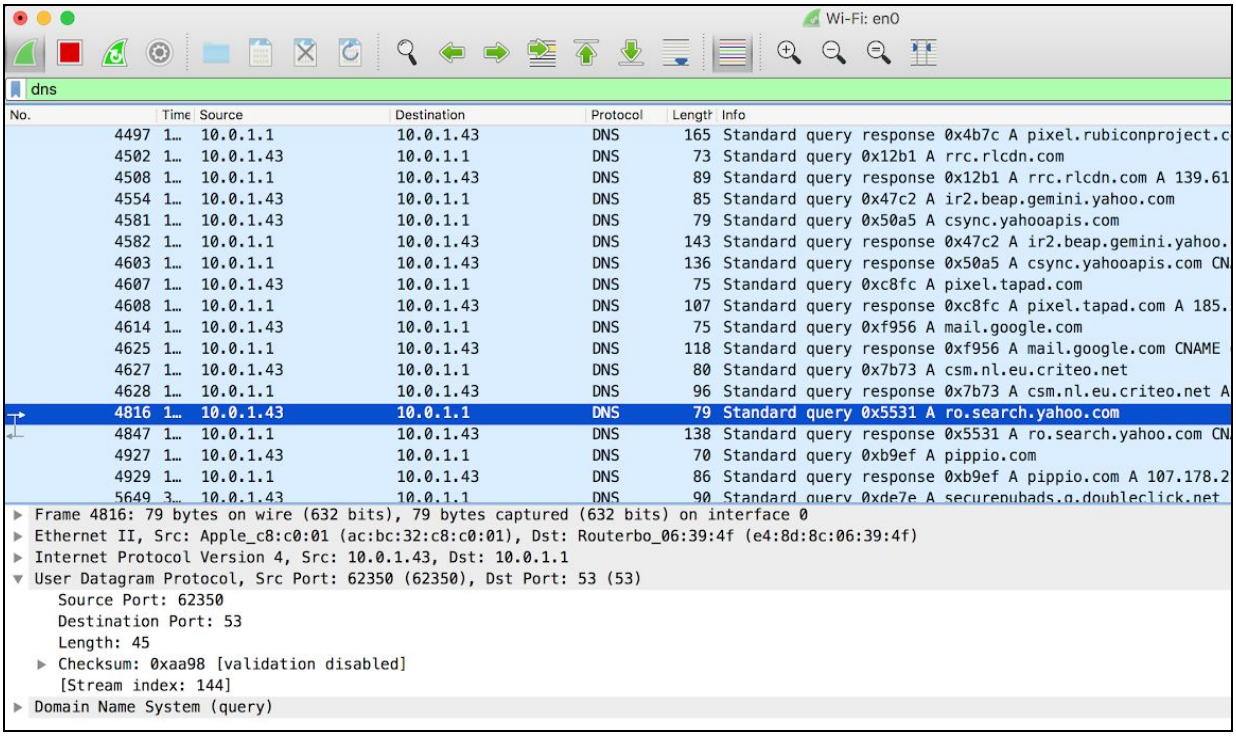


Figura 6.10

Acum sa luam un alt exemplu (cel din figura 6.10) in care putem vedea traficul/cererile DNS (de rezolvare de nume intr-o adresa IP).

Putem vedea ca protocolul DNS (despre care vom vorbi mai in detaliu in capitolul 7) foloseste UDP-ul pentru transportul datelor si mai exact foloseste portul 53 (un subiect pe care il vom aborda in cele ce urmeaza).

3) Porturi

Un port identifica in mod unic o aplicatie de retea (server Web, DNS etc.) pe un dispozitiv dintr-o retea. Fiecare port are un identificator – un numar care poate avea o valoare de la **1 – 65535**.

In momentul in care un PC trimite o cerere (pentru o pagina Web) catre un server, aceasta cerere va contine (printre altele) urmatoare informatii:

IP Sursa: PC

IP Destinatie: Server

Port Sursa: 29813 (generat random de catre Browser)

Port Destinatie: 80

Altfel spus, toate acestea reprezinta: Browser-ul (29813) PC-ului (sursa) cere pagina web (80) de la server (Destinatia).

Exemplu #1 - porturi TCP

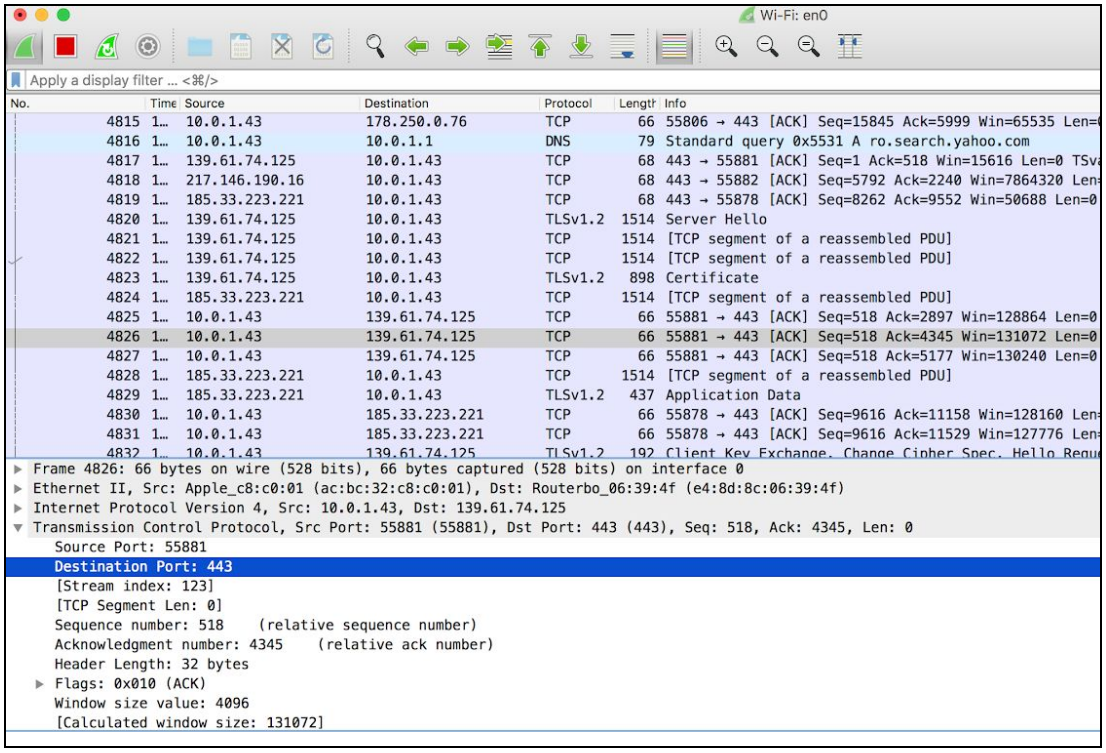


Figura 6.11

Acum haide sa luam cateva exemple in care sa putem analiza si vedea cele discutate mai sus. Dupa cum poti vedea in figura 6.11, exista un flux de comunicare intre 2 dispozitive (sursa: 10.0.1.43, destinatia: 139.61.74.125).

Portul sursa (generat aleator) in acest caz este **55881** (si cel mai probabil a fost generat de o aplicati **browser** - Google Chrome, Safari, Firefox etc.), iar **portul destinatie** este **443** (**HTTPS**, deci o aplicatie web securizata).

Astfel sursa se adreseaza unui server din Internet cerand pagina web gazduita de acest server. Pe langa asta, mai poti observa si o parte din header-ul TCP-ului despre care am vorbit la inceputul acestui capitol.

Exemplu #2 - porturi UDP

Tot in aceasta figura din Wireshark:

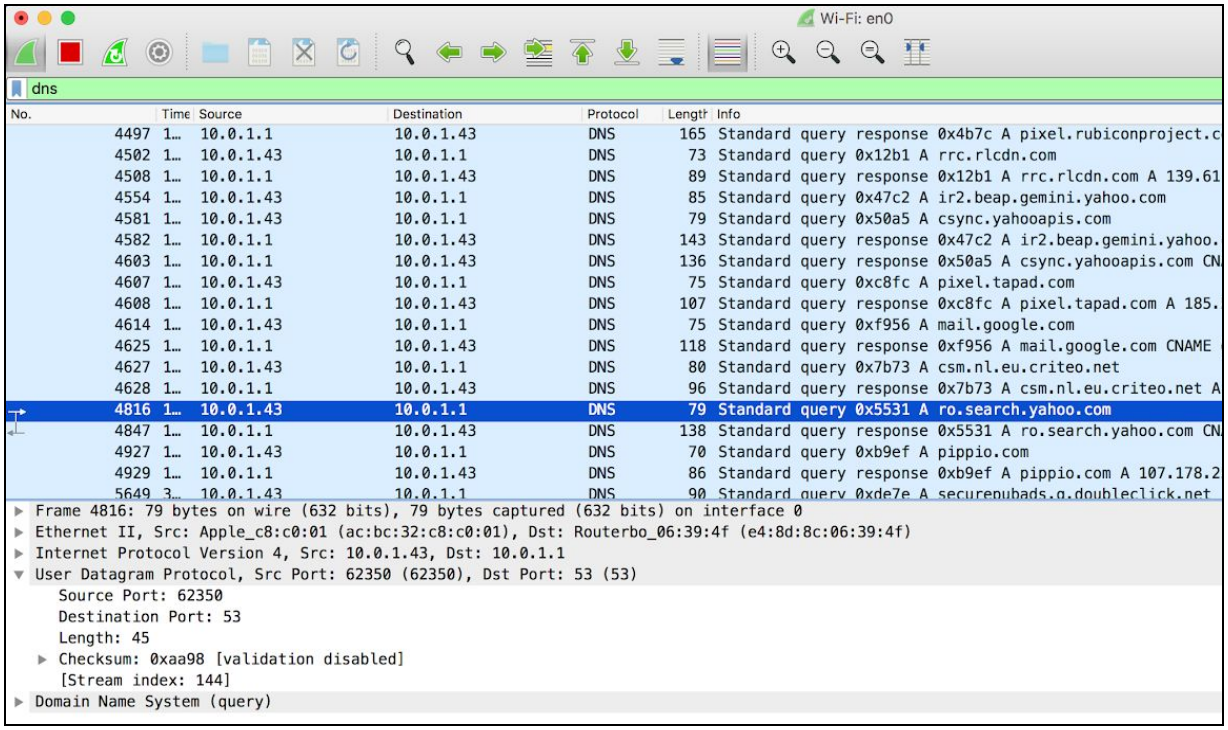


Figura 6.12

In exemplul din figura 6.12 poti vedea un proces similar cu cel anterior, dar in acest caz este vorba de **protocolul UDP** (mai degraba DNS care foloseste UDP). Aici poti vedea cele 2 **porturi** (**sursa** - 62350 si **destinatie** - 53) ale celor 2 dispozitive participante in comunicare (sursa: 10.0.1.43 si destinatia: 10.0.1.1).

Pe langa asta, te rog sa observi **simplitatea** *header-ului UDP* fata de *cel al TCP-ului* din figura 6.11. Cele mai importante elemente din header-ul UDP-ului sunt cele 2 porturi (cel sursa si cel destinatie).

Capitolul 7 - Nivelul 5, 6, 7 - Sesiune, Prezentare, Aplicatie

In acest capitol vorbim despre **ultimele 3 nivele** din modelul OSI si anume: **Sesiune**, **Prezentare** si **Aplicatie**.

a) Nivelul 5 - Sesiune

Scopul nivelului Sesiune este de a crea o sesiune, de a o mentine si de a o termina (atunci cand e cazul) intre 2 aplicatii de retea. O sesiune (comunicare) consta in schimbarea unui flux de date de tip cerere-raspuns intre 2 device-uri conectate la Internet. Device-ul care **cere datele** poate denumirea de **client**, iar device-ul care **ofera** datele se numeste **server**.

Un protocol important la acest nivel este **LDAP** (**L**ightweight **D**irectory **A**ccess **P**rotocol), un protocol care se ocupa cu gestionarea, cautarea si modificarea unui directory service (loc in care sunt stocate datele utilizatorilor din companie). Cu ajutorul acestui protocol putem obtine cateva elemente cheie in retea (mai exact in procesul de securizare al retelei):

- **Autentificarea** utilizatorilor in retea
- **Autorizarea** acestora

b) Nivelul 6 - Prezentare

Scopul nivelului Prezentare este de a "servii datele" intr-un anumit format (ex: JSON, formatul pozelor JPEG, PNG etc.). La acest nivel datele sunt structura sub o anumita forma si sunt livrate pentru a putea fi interpretate de aplicatile care ruleaza pe servere. Aceste date pot fi sub forma unui JSON (figura 7.1), XML (figura 7.2), etc.

```
"employees": [
    {
        "id": 52626,
        "name": "Employee One"
    },
    {
        "id": 26565,
        "name": "Employee Two"
    }
]
```

Figura 7.1

```
<SampleXML>
  <Colors>
    <Color1>White</Color1>
    <Color2>Blue</Color2>
    <Color3>Black</Color3>
    <Color4 Special="Light">Green</Color4>
    <Color5>Red</Color5>
  </Colors>
  <Fruits>
    <Fruits1>Apple</Fruits1>
    <Fruits2>Pineapple</Fruits2>
    <Fruits3>Grapes</Fruits3>
    <Fruits4>Melon</Fruits4>
  </Fruits>
</SampleXML>
```

Figura 7.2

Un alt exemplu pentru acest nivel este criptarea datelor. Datele criptate (aka. securizate) au scopul de a ascunde continutul initial intr-un alt format, aici intervenind nivelul Prezentare. In figura 7.3 poti vedea o captura in Wireshark a datelor criptate.

```
4 90 A3 31   .$V.sI.. ..^..E...<]..@......D2a^4......W04..1
F 6A 10 10   @...F.%......@.$L.N....U.....&.........B.@Oj..
6 D9 33 3C   ..=.+.".P<.&....}........o....T.@P.ni._.#...3<
0 73 75 2E   sl-.w.....E.4$s.`j.xrc4.[y.A....  0HT..d.a.su.
6 B0 78 D1   (F..T.X....d. ........j.....w.V=.+(h.s.E....x.
7 6B F4 D2   [.n.h!+...HUd]....T.1.......,F.BB P.>.n....k..
1 17 55 CA   ....:q6..hK.j.......].Z................R.zA.U.
1 D5 00 9C   ..>..5[..E.E..n.....*7.B.g+..]+.`.]..X........
5 FF ED D0   KI.O.H........*.6..`......bE.P_.|.............
9 77 A0 B0   .....5*...8.+o.< w.GvA.V*.zH....g....F<.]..w..
E A3 F1 88   ....+.?wsP7.,..k\.`.....TE..'X{...m.a....s....
C 92 C1 C3   .Z,z..vw^.!....r.Q.mh.C.d..g.GMk...T.X........
7 10 4E 71   \k...F....]FD....#.9.................%....]..Nq
7 23 93 9B   .........I&N...b..[R.9.BC.!.fR.7..{..2V._{:.#..
F DC B9 F0   .....z..<.JP....r@.........~...3.3..S..&......
F 7A 16 27   ...(,~M.........y:....le........N\.....k.._z.'
C FE 46 BE   .Wl.u..!o.....5.:LIE..5)+VH....H.^.G.....T..F.
7 18 FD 05   ....2b...FV.E..^..I..3.R.....Q..^`..g}:"f.g...
E B1 CB 2A   v+!..>.].tAp.@........x._..VN..W5.....46.H...*
E 11 78 6D   =.L.\.<...8x....!..\.@..Y.3..#)OJ..'.=.`E...xm
F 0C 17 CE   .e.Cv..X.X..RQ@.oj.c...hp...b..,{...2.&).. ...
D E7 8D FB   X.1..J.eI.6n.......i....=F...*;Zth.. ..^X.....
1 C0 B0 C1   <......Pq"T!......fyw..<.."...f(z.S...........
E 17 8D BF   ...9...}...4... S.+.\6.......8.V.....7pK.z....
```

Figura 7.3

c) Nivelul 7 - Aplicatie

Cand vorbim de aplicatii, in acest context, ne referim strict la **aplicatii de retea.** Aceste aplicatii de retea sunt, in general, cele oferite de catre un server (ex: *aplicatie Web, de email, acces remote la un PC*, etc.). Iata cateva protocoale care functioneaza la acest nivel:

- **HTTPS** (portul 443 TCP)
- **SSH** (portul 22 TCP)
- **DHCP** (portul 67/68 UDP)
- **DNS** (portul 53 UDP)

De exemplu **HTTPS** (Hyper Text Transfer Protocol Secure) este un protocol care ne ajuta sa accesam site-urile web intr-un mod securizat. **HTTP** este varianta nesecurizata care ofera doar functionalitatea (posibilitatea accesarii paginilor web). Iata mai in figura 7.4 un exemplu:

Figura 7.4

Un alt protocol important care ne ajuta sa accesam site-urile din Internet este **DNS** (**D**omain **N**ame **S**ervices) care tranforma un nume de domeniu (ex: www.google.ro) intr-o adresa IP (pentru ca toate echipamentele de retea FOLOSESC adresele IP si nu numele de domeniu).

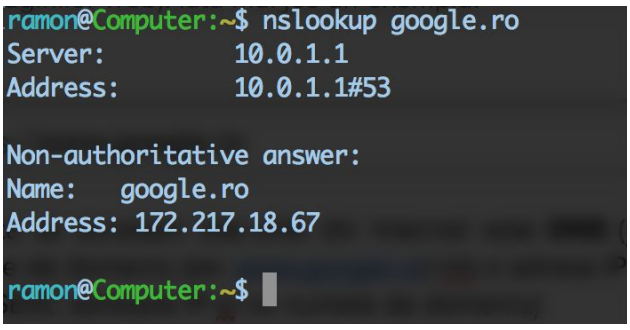


Figura 7.5

In figura 7.5 poti vedea adresa IP a celor de la Google: **172.217.18.67**, folosind comanda **nslookup.**

Aplicatii de Retea

Iata cateva protocoale des intalnite ale aplicatiilor de retea:

HTTP

- **Descriere:** folosit pentru traficul Web (transporta fisierele HTML de la server la client)
- **Port:** 80
- **Protocol de Transport:** TCP

HTTPS

- **Descriere:** folosit pentru traficul Web intr-un mod **securizat**
- **Port:** 443
- **Protocol de Transport:** TCP

FTP

- **Descriere:** permite transferul de fisiere intre un client si un server
- **Port:** 20/21
- **Protocol de Transport:** TCP

DNS

- **Descriere:** gaseste IP-ul unui nume de domeniu (ex: *google.ro -> 172.217.18.67*)
- **Port:** 53
- **Protocol de Transport:** UDP (client), TCP (server)

Telnet

- **Descriere:** permite conexiunea **nesecurizata** de la distanta catre un echipament (Switch, Router)
- **Port:** 23
- **Protocol de Transport:** TCP

SSH

- **Descriere:** permite conexiunea **securizata** de la distanta catre un echipament (Switch, Router)
- **Port:** 22
- **Protocol de Transport:** TCP

DHCP

- **Descriere:** aloca in mod dinamic adrese IP, masca si default gateway device-urilor din retea
- **Port:** 67/68

- **Protocol de Transport:** UDP

SMTP

- **Descriere:** protocol de transfer de mail-uri intre serverele de mail
- **Port:** 25
- **Protocol de Transport:** TCP

IMAP

- **Descriere:** protocol de transfer de mail-uri de la server catre client (mail-urile vor fi stocate pe server)
- **Port:** 143
- **Protocol de Transport:** TCP

POP3

- **Descriere:** transfera mail-uri de la server la client (si le stocheaza pe PC-ul acestuia)
- **Port:** 110
- **Protocol de Transport:** TCP

RDP

- **Descriere:** permite conectarea de la distanta (in mod grafic - GUI) la o masina Windows, Linux sau MacOS
- **Port:** 3389
- **Protocol de Transport:** UDP

Acum, propun sa luam cateva dintre protocoalele discutate mai sus si sa vorbim putin mai in detaliu despre acestea.

1) Telnet

Telnet este un protocol de retea care permite conectarea de la distanta la un echipament de retea (Router, Switch, Firewall, etc.) sau la un server.
Este foarte raspandit, aflandu-se, instalat, pe marea majoritate a device-urilor, dar are un **Dezavantaj** major: Conexiunea este **NESECURIZATA** ! Adica, toata comunicatia dintre 2 device-uri folosind Telnet, este trimisa in **clear text** si poate fi foarte usor interceptata de catre un atacator.

Un program pe care il putem folosi pe Windows este PuTTY (figura 7.11). Cu el ne putem conecta la un dispozitiv prin *Telnet*, *SSH* sau chiar prin *consola* (Serial).

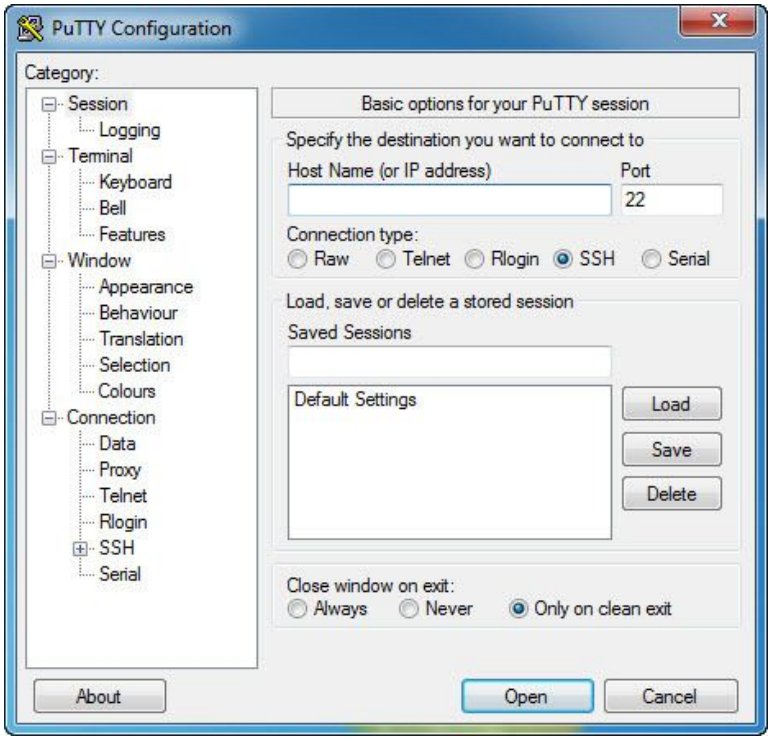


Figura 7.11

Telnet foloseste **portul 23 pe TCP** si se poate porni pe Linux, Mac, Windows sau orice alt dispozitiv, dar recomandarea este sa nu-l folosim (datorita faptului ca mesajele nu sunt criptate, ci transmise in clear text).

2) Secure Shell (SSH)

SSH este un protocol de retea care **permite conectarea de la distanta** la un echipament de retea (Router, Switch, Firewall, etc.) sau la un server in **MOD SECURIZAT**. Acesta este **cel mai folosit protocol** pentru conexiunile de la distanta (remote) datorita securitatii si flexibilitatii pe care o ofera. SSH foloseste **portul 22 pe TCP**.

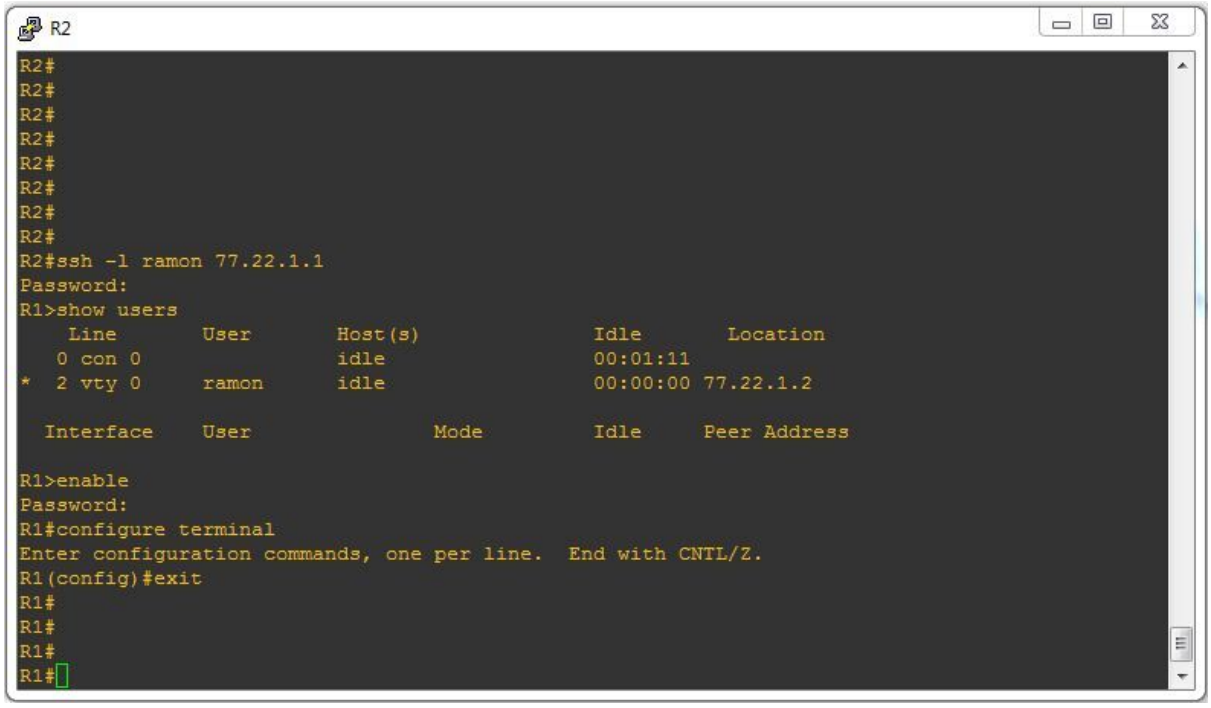


Figura 7.12

3) RDP - Remote Desktop Protocol

Pentru ca am vorbit despre 2 protocoale care ne permit accesul de la distanta pe un echipament de retea sau server (prin CLI - linia de comanda), propun sa aruncam o privire si peste un protocol care ne ofera acelasi lucru (accesul de la distanta pe un alt calculator), dar vine cu interfata grafica in joc (GUI).

Iata un exemplu pentru protocolul RDP, un caz in care m-am conectat (de la distanta) la desktop-ul meu din alta locatie.

Practic (pe Windows) avem la dispozitie o astfel de fereastra (figura 7.13) in care vom introduce adresa IP (sau numele) PC-ului impreuna cu datele logare (user + parola):

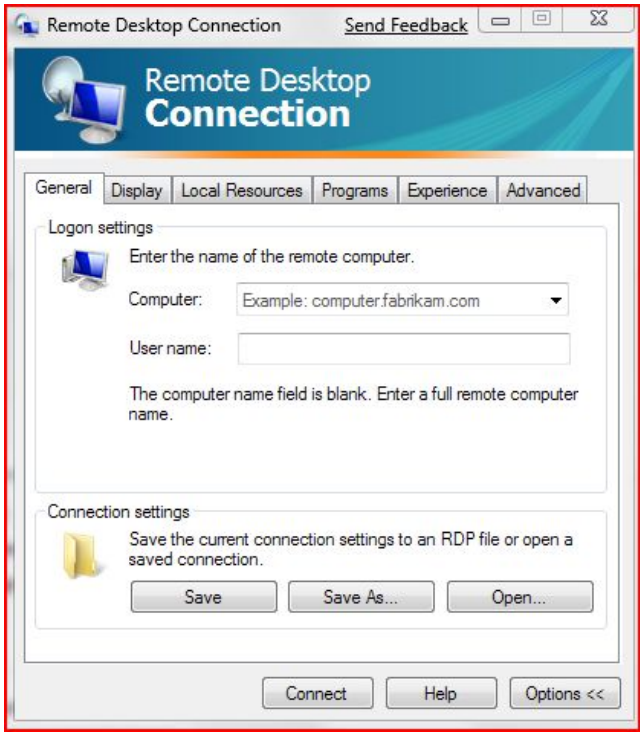


Figura 7.13

Iar in figura 7.14 putem vedea cum arata o astfel de conexiune (o simpla fereastra prin care suntem conectati la calculatorul specificat mai devreme).

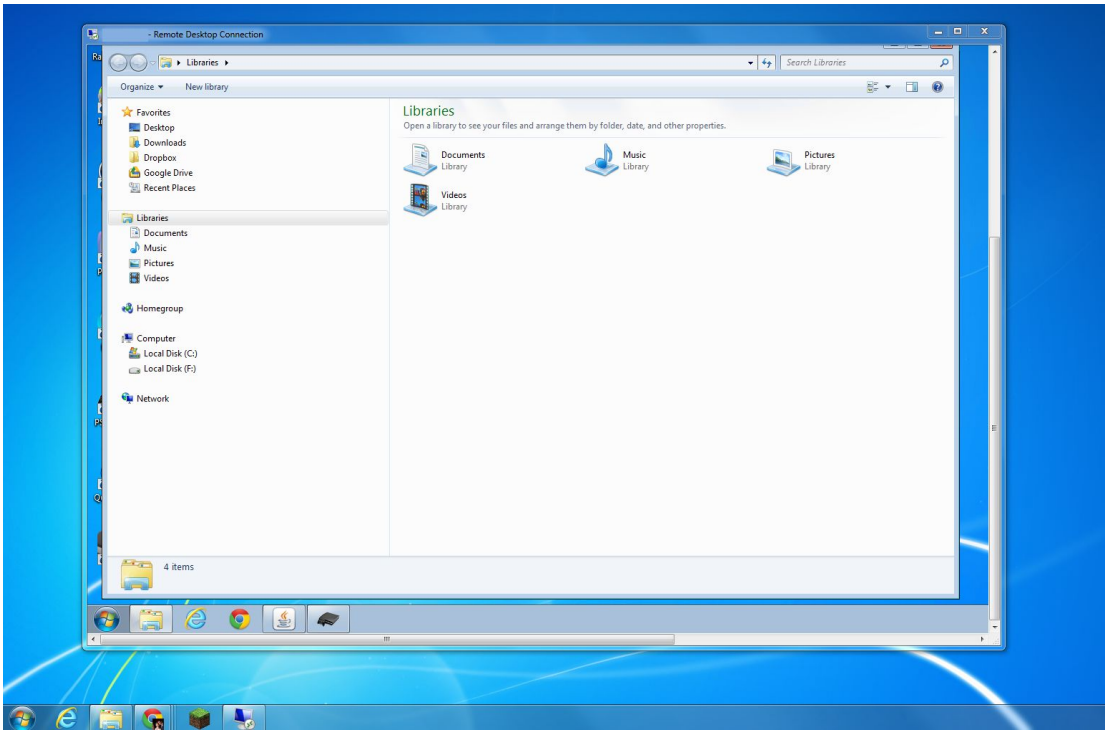


Figura 7.14

Capitolul 8 - Cisco IOS & Introducere in CLI

Cand vine vorba de echipamentele Cisco (Routere si Switch-uri) toate au un lucru comun: Sistemul de Operare, **Cisco IOS.** IOS vine de la **I**nternetwork **O**perating **S**ystem si este “motorul” care ofera puterea acestor echipamente. Noi putem lucra cu IOS in 2 moduri **CLI** sau **GUI**. Acronimul **CLI** reprezinta **C**ommand **L**ine **I**nterface si este linia de comanda in care vom configura toate aceste dispozitive. Astfel vom avea **acces total** asupra Router-ului sau a Switch-ului si ii vom putea influenta comportamentul in retea.
GUI reprezinta **G**raphical **U**ser **I**nterface si cel mai des il folosim atunci cand configuram echipamente de retea mici precum Routerele Wireless (Wi-Fi) de acasa.

In cele ce urmeaza vom vedea cum putem interactiona cu acest IOS prin **CLI** si vom folosi versiunea **15.2** sau **12.4**.

Introducere in CLI - Configurari de Baza

In aceasta sectiune vom trece la partea practica si anume vom incepe cu setarile de baza pe Routere Cisco. Pentru inceput iti recomand sa folosesti simulatorul **Cisco Packet Tracer** versiunea **6.2** pe care il gaseti **AICI**.

Configuratile de baza includ urmatoarele:

- Numele dispozitivului (Hostname)
- Parole (criptate sau in clear text)
- Adrese IP pe interfete
- Setare acces remote prin Telnet sau SSH

a) Nivele de Acces

In Cisco IOS avem mai multe nivele de acces, in care, utilizatorul poate *face teste de conectivate* (>) folosind comenzi precum *ping* sau *traceroute*, ***vedea*** anumite setari (#), poate ***schimba setarile*** - (config)#.

Pentru a trece dintr-un nivel de acces in altul trebuie sa dam fie comanda **enable,** fie comanda **configure terminal**.In momentul in care te conectezi la un echipament de retea Cisco (Router, Switch, Firewall etc.) te vei afla in ***user exec mode (>).***

In acest mod (***user exec***) esti ***limitat*** *din punct de vedere al comenzilor* pe care le poti da pe echipament (in general comenzi precum ping, traceroute, etc.) Pentru a trece in urmatorul nivel de acces (in care ai mai multe privilegii) trebuie sa introduci comanda:

*Router>**enable***
In urma aceste comenzi te vei afla in nivel **priviledged mode - R1#.** Aici poti sa vezi tot ce se intampla pe echipament (prin comenzi de **show**), dar **NU poti face modificari**.

*Router>**enable***
Router#

Pentru a putea face modificari pe echipament trebuie sa treci intr-un nivel de acces cu si mai multe privilegii si anume **global configuration mode**:

*Router#**configure terminal***
Router(config)#

Acum poti face **orice modificare** doresti pe aceste echipamente de retea. Acest *global configuration mode* este echivalentul **Administrator**-ului din Windows sau **root**-ului de pe sistemele Linux. Iata toate comenziile cu care este recomandat sa fii familiar:

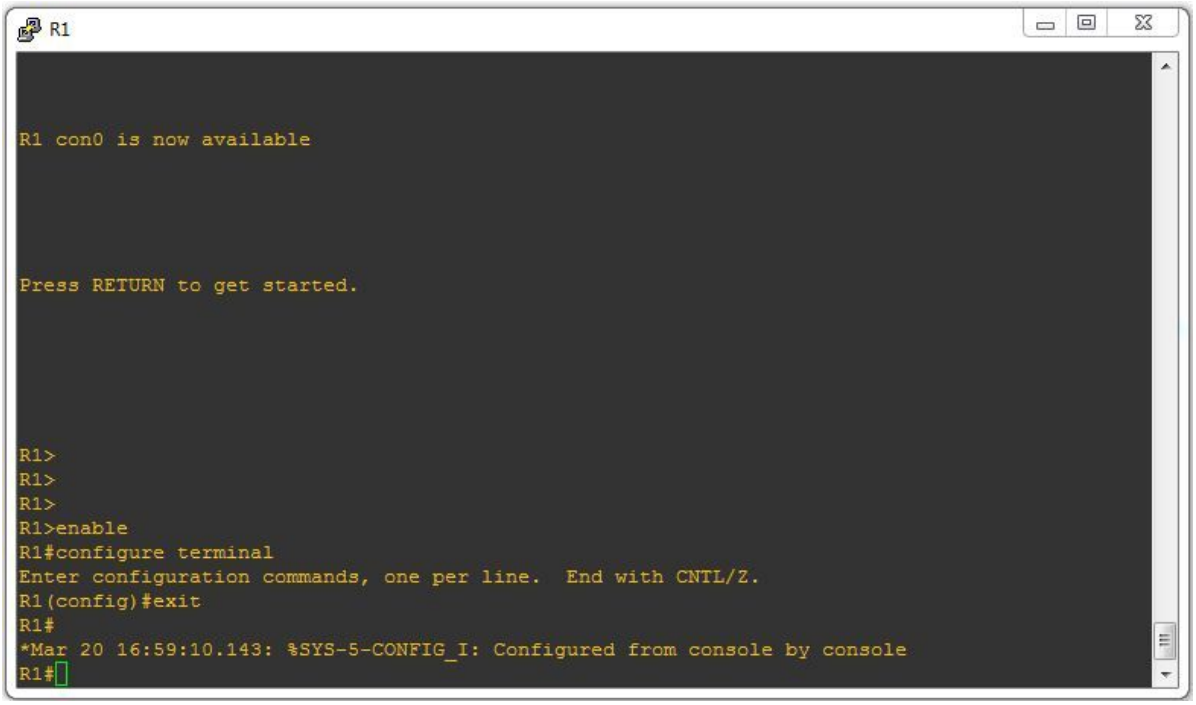


Figura 8.1

Daca vrem sa scriem o comanda mai lunga (si nu avem chef sa o tastam :D) avem o solutie care garanteaza scrierea acesteia mai rapida (si corecta).

Daca scriu comanda R1#**show run**, si apas tasta **TAB**, va face **autocomplete** comenzii. De asemenea "**?**" ne va afisa comenzile (urmatoare) disponibile.

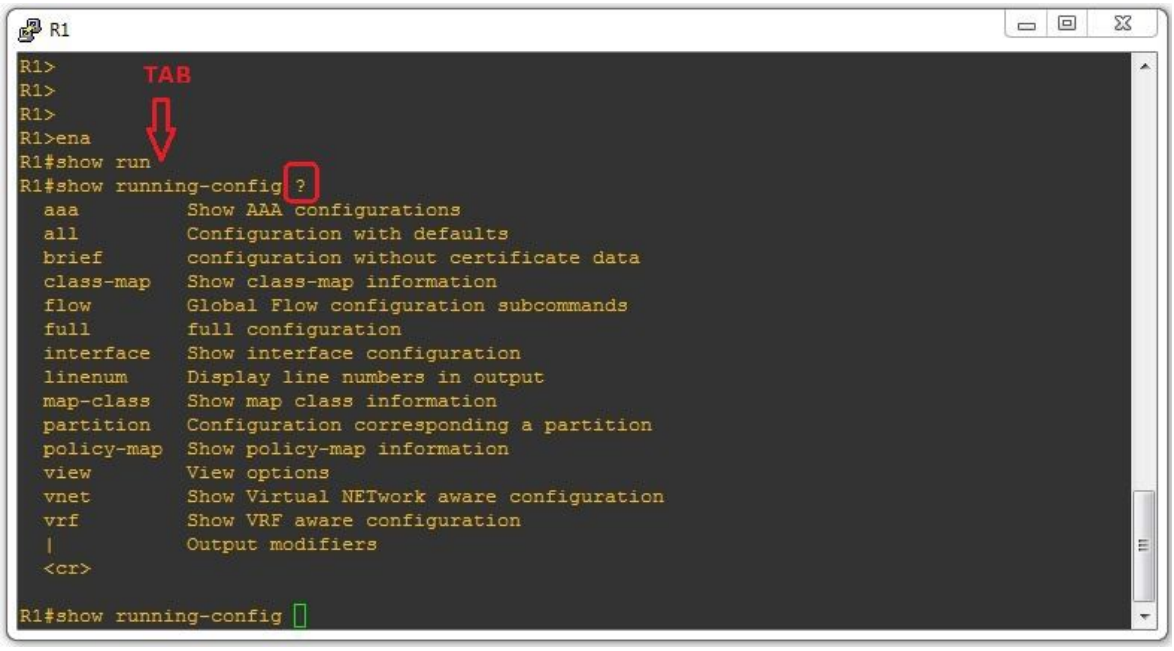


Figura 8.2

b) Setarea numelui unui dispozitiv (Hostname)

In exemplul de mai sus, pentru a schimba numele Router-ului (sau al Switch-ului) trebuie sa introducem urmatoarea comanda:

```
Router(config)#hostname NUME_ROUTER
NUME_ROUTER(config)#
```

c) Securizarea accesului pe Router

Acum sa vedem cum putem securiza accesul pe Router prin setarea unei parole. Setarea unei **parola** la nivelul **priviledged mode** (*#*):

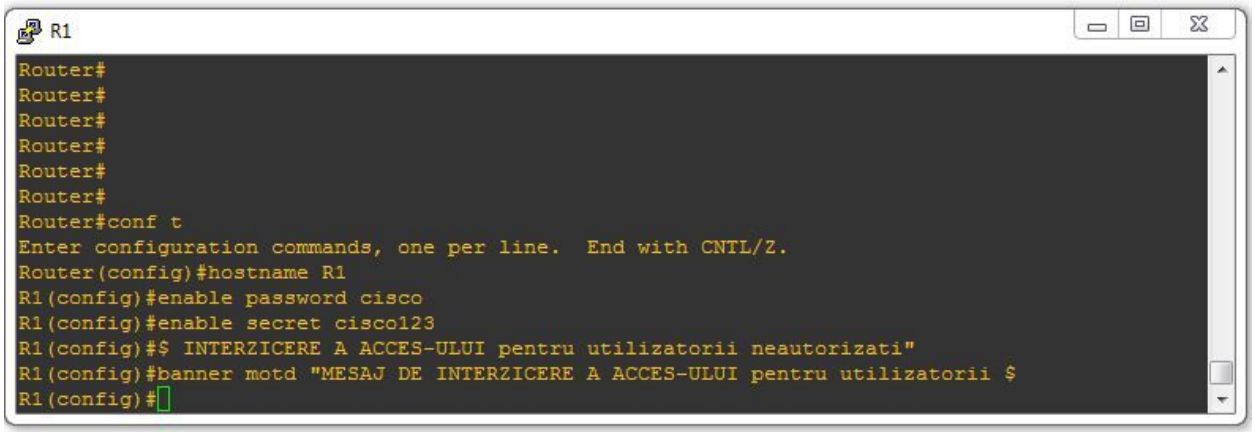


Figura 8.3

`Router(config)#hostname R1`

`R1(config)#enable password cisco`
sau
`R1(config)#enable secret cisco`

Setarea unui **banner** de atentionare la *logarea* pe echipament:
`R1(config)#banner motd "MESAJ DE INTERZICERE A ACCES-ULUI pentru utilizatorii neautorizati"`

Poate te intrebi care este diferenta intre *enable password* si *enable secret* ? Ei bine, iata diferenta (figura 8.4):

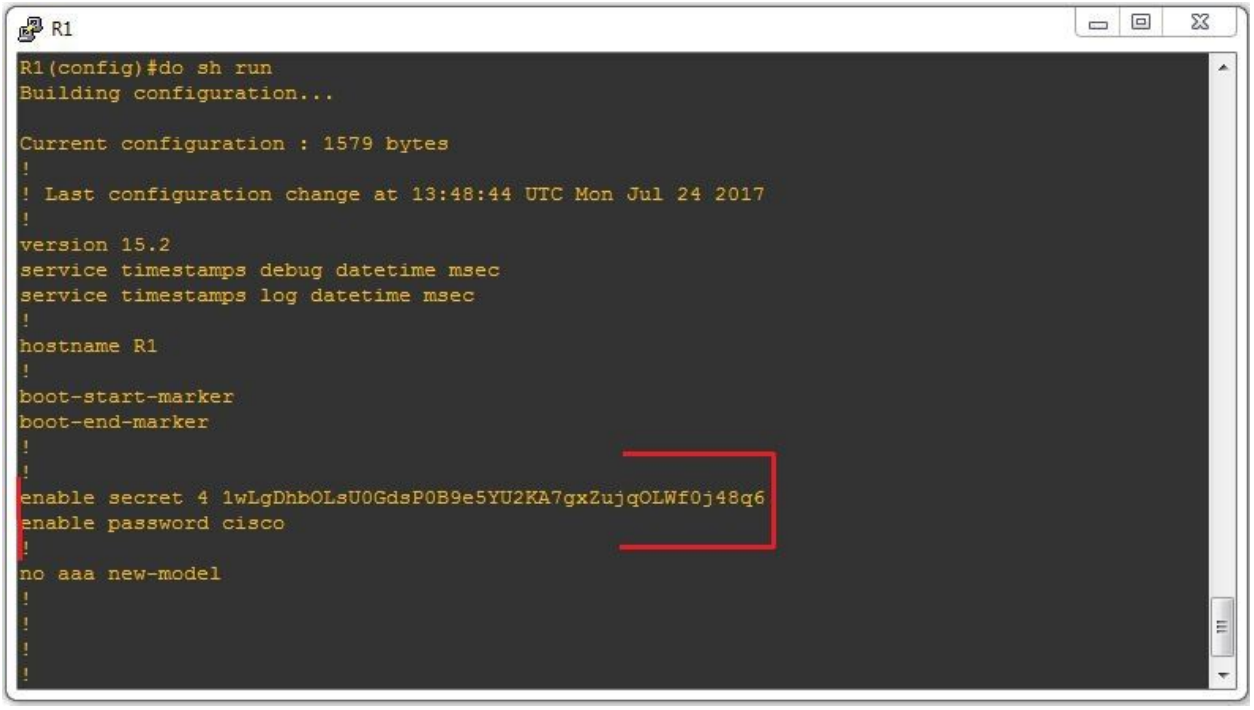
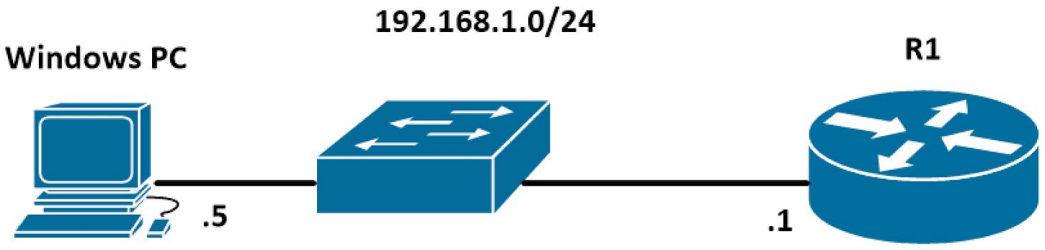


Figura 8.4

Dupa cum poti vedea una este stocata intr-un **mod criptat** (*#enable secret*), iar cealalta este stocata in **clear text** (*#enable password*).

Haide sa aruncam o privire la topologia de mai jos si sa incepem sa configuram Routerul pentru accesul la retea:



d) Setarea unei adrese IP pe Router

Routerul **interconecteaza** mai multe retele prin **porturi** (de obicei 2 - 3). Denumirea de **porturi** este valabila cand ne referim din punct de vedere **fizic**. Cealalta denumire este de **interfeta** si este valabila cand ne referim din punct de vedere **logic** (mai exact ceea ce vom configura noi, putin mai tarziu).

Port = Fizic
Interfata = Logic

De exemplu: "*noi vom seta o adresa IP (logic) pe o interfata si vom conecta cablul (fizic) intr-un port*"

Aceste interfete trebuie sa aiba *configurata o adresa IP* pentru a putea comunica cu reteaua si interfata trebuie sa fie ***PORNITA***. Setarea unei adrese IP pe o interfata se face in felul urmator:

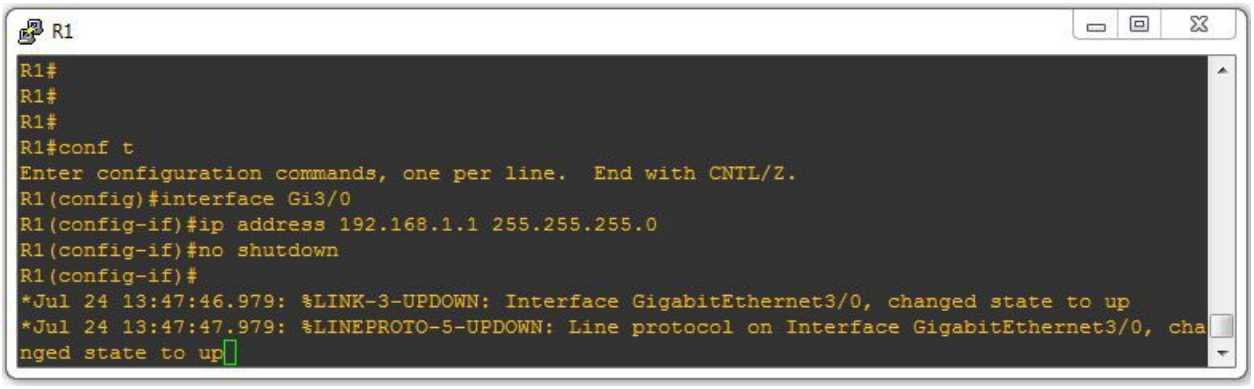


Figura 8.5

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
```

f) Configurare access remote pe Router (Telnet, SSH)

Putin mai devreme in acest capitol am invatat ce este Telnet, respectiv SSH, acum a sosit momentul sa le si configuram pe Router:

Telnet

Figura 8.6

```
R1(config)#line vty 0 14
R1(config-line)#password cisco
R1(config-line)#login
```

Vom intra pe linile virtuale (15 in total), vom seta parola “cisco” si vom porni procesul (Telnet) prin comanda #login.

SSH

Dupa cum am vorbit si in capitolul 7, SSH-ul este un protocol care ne asigura conectarea remote catre un echipament din LAN sau din Internet, intr-un mod securizat. **Pentru a configura SSH** pe un echipament Cisco vom parcurge urmatorii pasi:

1. *Crearea unui user & parola*
2. *Nume de domeniu*
3. *Pereche de chei publica si privata*
4. *Pornirea procesului pe linile virtuale (vty) prin #login local*

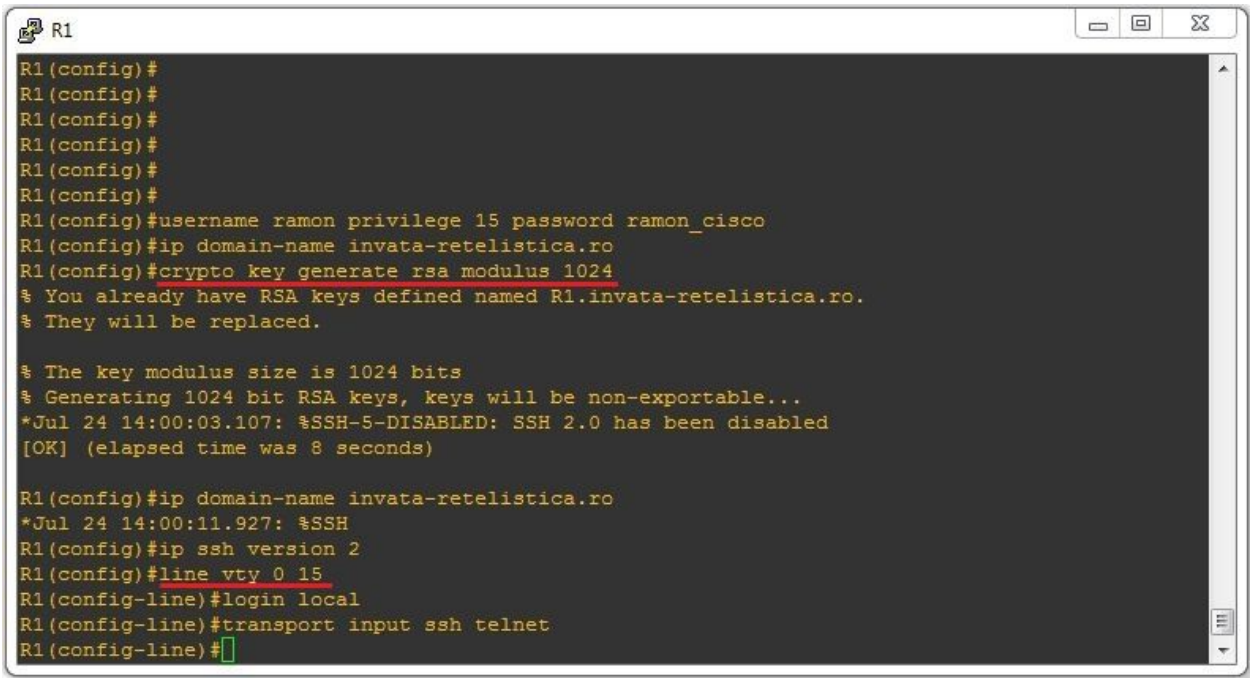


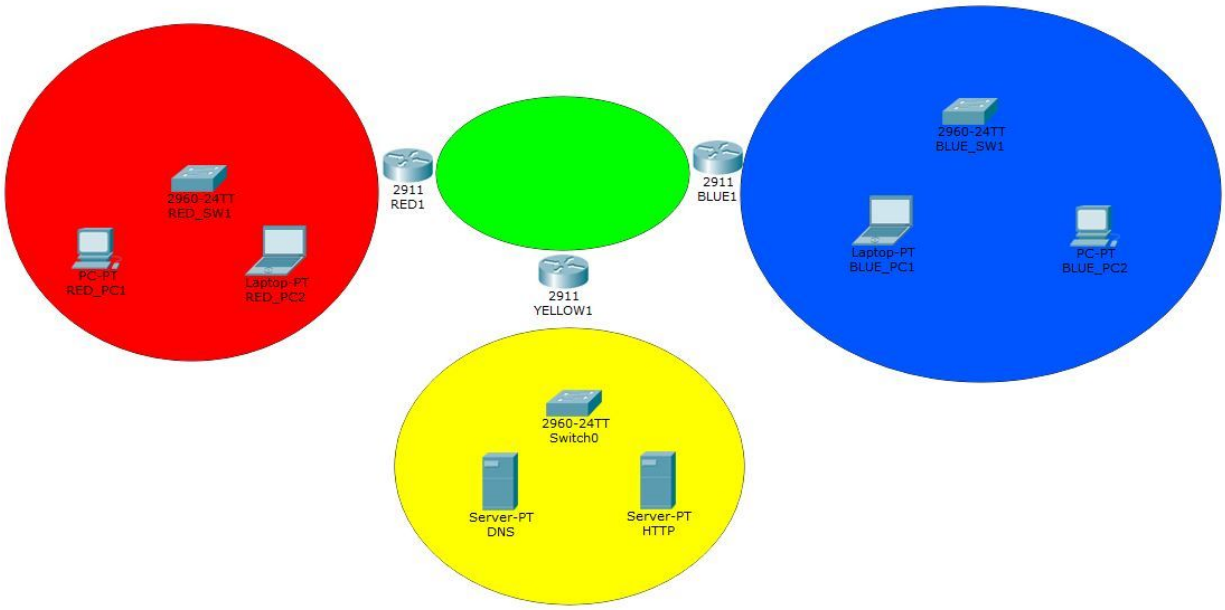
Figura 8.7

```
R1(config)#username ramon privilege 15 password ramon_cisco

R1(config)#ip domain-name invata-retelistica.ro
R1(config)#crypto key generate rsa modulus 1024
R1(config)#ip ssh version 2

R1(config)line vty 0 15
R1(config-line)#login local
R1(config-line)#transport input ssh telnet
```

Laborator #1

Acum am ajuns la partea de laborator (partea practica). Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele de mai jos si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Acomodarea cu CLI. Configurari de baza pe Routere si Switch-uri. Asigurarea conectivitatii



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Cerinte laborator

0) Cablati corespunzator echipamentele (atentie la tipurile de cabluri folosite)

1) Setati adrese IP pe Routere, Switch-uri si PC-uri

- **RED**: 10.16.22.0/24
- **GREEN**: 89.12.0.0/24 (impartiti reteaua in retele a cate maxim 2 IP-uri disponibile pentru Routere)
- **BLUE**: 192.168.0.64/27
- **YELLOW**: 172.30.33.128/25

Alocati prima adresa IP Routerelor, urmatoarele adrese IP PC-urilor din retea si ultima adresa IP din fiecare retea, alocati-o Switch-urilor.

2.1) Setati Hostname, Parole de Enable

2.2) Setati un Banner pe cele 2 Routere:

("*DOAR ACCESUL AUTORIZAT*")

3.1) Configurati Telnet pe RED1, RED_SW1, BLUE_SW1

- Folositi parola: *secrePass*

3.2) Configurati SSH pe BLUE1 si YELLOW1

- Folositi ce username si parola doriti

4) Asigurati conectivitate end-to-end intre retele (folosind Rute Statice)

5) Testati conectivitatea prin comanda ping, intre :

- PC-urile din aceeasi retea (RED, BLUE)
- PC-uri si Routere
- PC-uri din retele opuse
- PC-uri si Servere (accesati IP-ul serverelor din Browser)

Descarca folderul cu laboratoare de AICI.

Capitolul 9 - Concepte de Baza despre Rutare

1) Cum Functioneaza un Router ?

In primul ce este un Router ? Este nimic mai multe decat un calculator. Are aceleasi caracteristici cu acesta:

- **Procesor** - CPU
- **Memorie RAM** (128MB+ depinde de model), ROM
- **Spatiu de stocare** in Flash - 32MB+ depinde de model
- **Sistem de Operare** - Cisco IOS

Toate aceste componente hardware propulseaza creierul fiecarui echipament de retea (Router, Switch, Firewall etc.) si anume Sistemul de Operare (OS) al celor de la Cisco - IOS. IOS vine de la **Internetwork Operating System**. Acest OS ii ofera “puterea” Routerului si face diferenta intre Cisco si celelalte branduri din lumea Retelelor.

Cisco, in momentul de fata, este lider mondial pe piata echipamentelor de retea. Ei ofera diferite solutii precum: Routing, Switching, Security (Firewall-uri, IPS, SpamFiltere etc.) la Voce, Video, Data Center si asa mai departe. Noi (in mare) ne vom concentra pe elemente legate de partea de Routing.

Routerul avand in componenta elementele descrise anterior trebuie sa treaca printr-un proces pentru a deveni operational si pregatit sa-si faca treaba. Acest proces de pornire (bootare) arata asa:

1. **POST** (**P**ower-**O**n **S**elf **T**est): test asupra componentelor Hardware (CPU, RAM, etc.)
2. **Bootstrap**: stabileste locul in care se afla OS-ul (Retea - TFTP sau Flash)
3. **Incarca imaginea OS-ul in RAM**:
4. **Incarca Fisierul de configurare**: *startup-config*

Toate echipamentele de retea au un fisier de configuratie (numit startup-config) in care sunt salvate setariile echipamentului. Acest startup-config se afla intr-o memorie speciala numita **NVRAM** (Non-volatile RAM). Aceasta memorie este una mica (< **64kB**) si nu isi sterge continutul cand se ia curentul (se opreste Routerul sau se restarteaza).

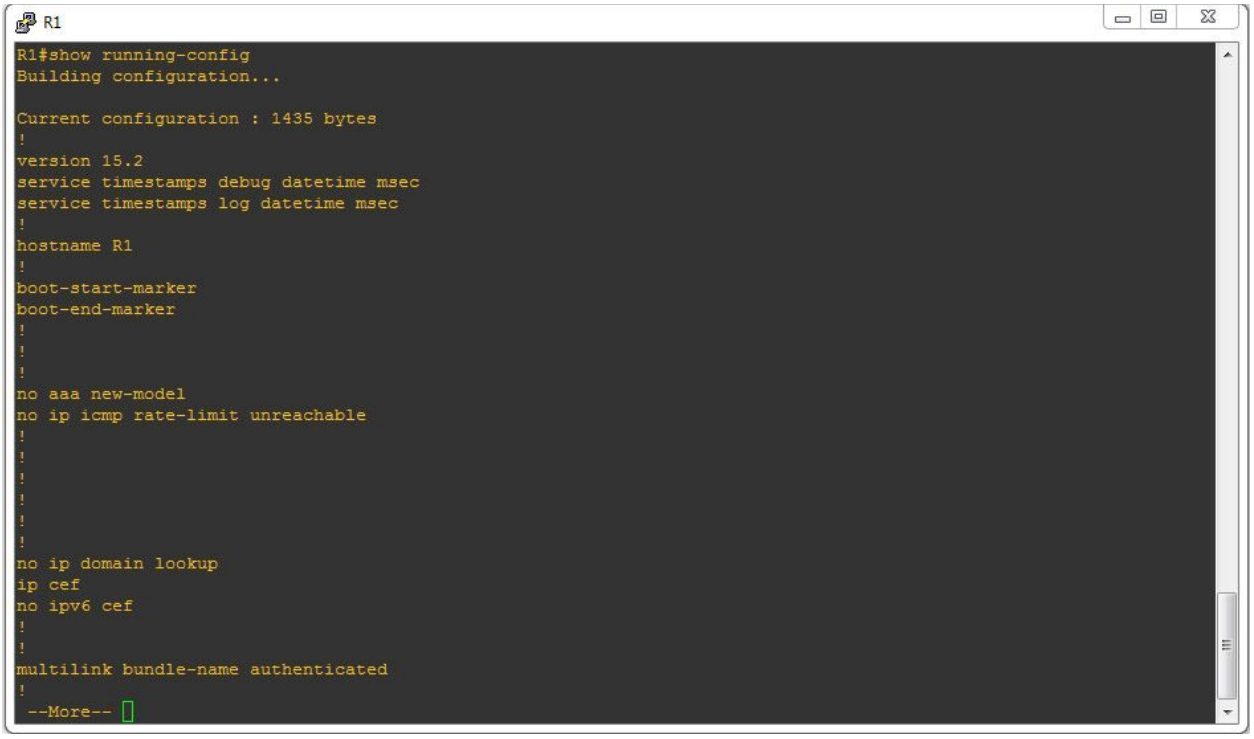


Figura 9.1

Odata ce porneste echipamentul, acest startup-config este copiat in RAM, intr-un fisier actual de configurare numit **running-config**. Acest fisier va contine setarile initiale si cele adaugate de noi pe parcurs ce echipamentul functioneza.

Orice modificare facem este scrisa in running-config. Odata ce am salvat modificarile facute, acestea vor fi scrise in startup-config !

Daca nu salvam modificarile, in momentul in care echipamentul va fi oprit (pierdere de curent, restart, etc.) acestea vor fi pierdute.

2) Tabela de Rutare

Ce reprezinta aceasta table de rutare ? Este locul in care un Router stocheaza informatia despre diferite retele. Este elementul fundamentul pe care il foloseste acest echipament de retea. Fara existenta ei, nu ar putea avea loc procesul de rutare (trimitere a pachetelor dintr-o retea in alta).

O **tabela de rutare** arata astfel (vezi comanda *#sh ip route*):

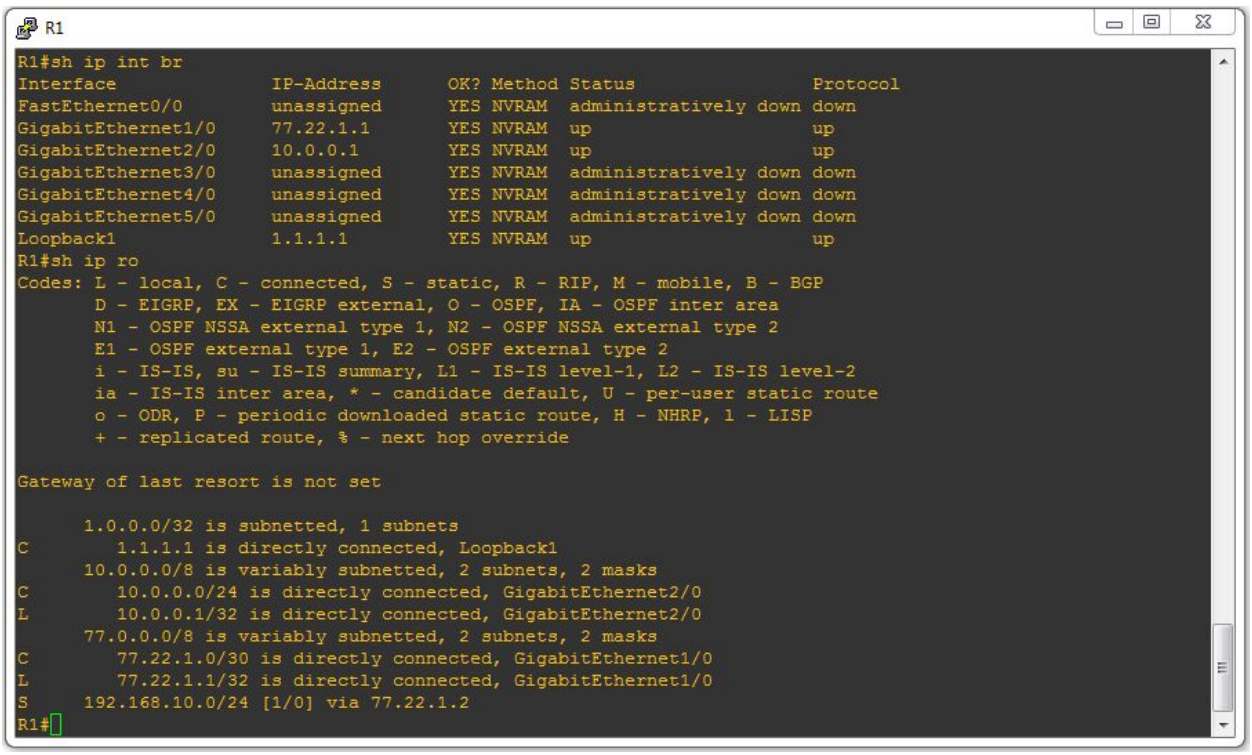


Figura 9.2

Cum functioneaza Rutarea pachetelor ?

Scopul unui Router este sa ruteze pachete. Mai exact acesta trebuie sa ia o decizie logica (pe baza tabelei de rutare) pe ce interfata trebuie sa trimita pachetul.

"Routerul primeste trafic pe o interfata si trebuie sa decida pe ce interfata il va trimite."

Tipul rutelor

Exista mai multe tipuri de rute:

- *Direct Conectate* - **C**
- *Statice* - **S**
- *Protocoale Dinamice de rutare* - RIP (**R**), OSPF (**O**), EIGRP (**D**), BGP (**B**)

La inceput (dupa ce a pornit) un Router stie doar de retele **direct conectate**. Acestea vor aparea in tabela de rutare (*#show ip route*) cu litera **C** in fata (asa cum poti vedea si in figura de mai jos). Litera **L** vine de la **Local** si reprezinta adresa IP a interfetei unui Router.

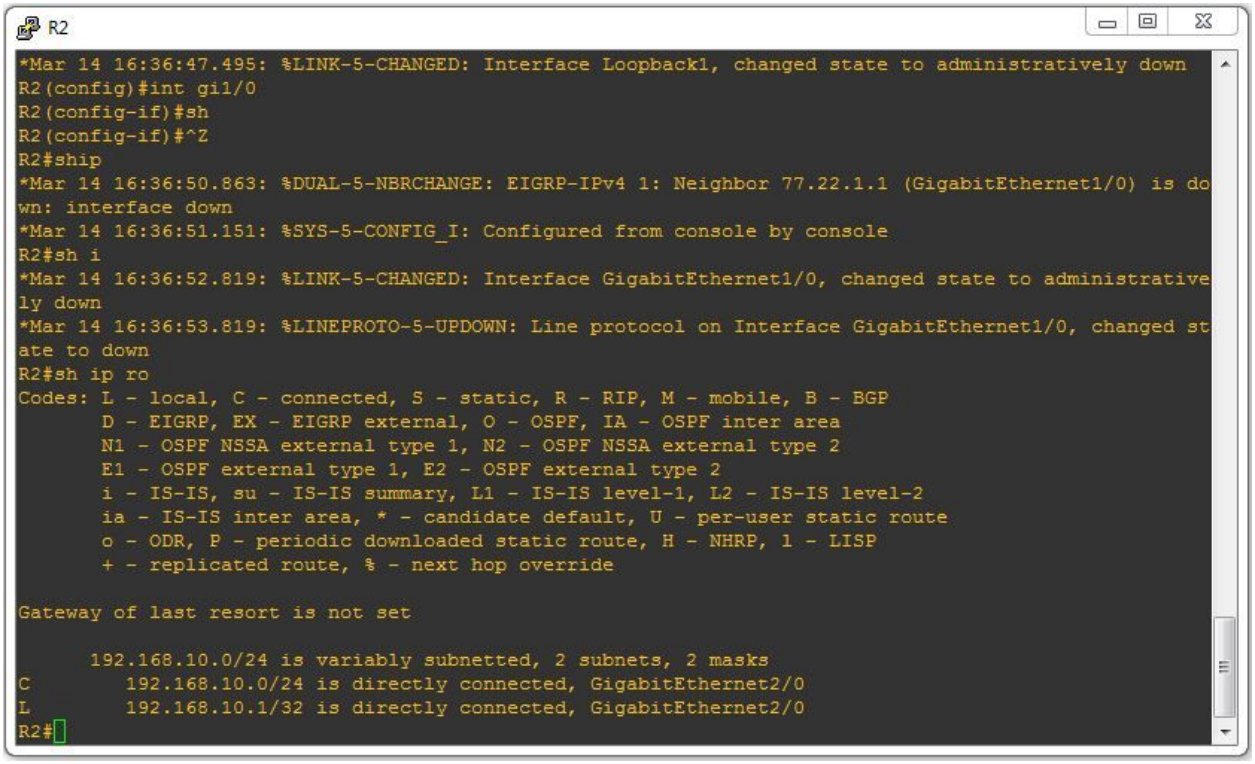


Figura 9.3

Dupa cum poti sa vezi reteaua direct conectata este 192.168.10.0/24 si se afla pe interfata GigabitEthernet 2/0. Adresa IP a Routerului de pe aceasta interfata este 192.168.10.1.

Distanta Administrativa si Metrica

Gandeste-te la o informatie (stire) pe care ai auzit-o de la un prieten foarte apropiat vs o persoana pe care abia ai intalnit-o pe strada. In cine vei avea mai multa incredere ? Clar ca in prieten. In acest scenariu exista 2 surse de informare diferite (pritenul si strainul). Nivel tau de incredere va fi mai mare in prieten fata de strain.

Ei bine, la fel functioneaza si un Router cand vine vorba de sursa informatiei. Acest concept se numeste Administrative Distance (AD - distanta administrativa) si este o valoare intre **0 - 255.** Un **AD mai mic** va insemna intotdeauna un ***nivel de incredere mai mare*** in acea ruta.

*“**AD** = cea mai credibila sursa”*
*“**Metrica** = cea mai buna cale spre destinatie“*

Fiecare tip de ruta are un astfel de AD. Exemplu:

Tip Ruta	**C** (Conectat)	**S** (Static)	**D** (EIGRP)	**O** (OSPF)	**R** (RIP)
AD	**0**	**1**	**90**	**110**	**120**

Metrica reprezinta factorul (numarul) care determina cea mai buna cale spre destinatie, care provine de la aceeasi sursa.

Spre exemplu, Google Maps iti poate da mai multe sugestii pentru a ajunge de la o anumita sursa (ex: Arad) catre o anumita destinatie (ex: Bucuresti). Poti ajunge la Bucuresti prin Sibiu sau prin Craiova, dar care va fi cea mai rapida cale ? Ei bine, aici intervine metrica, care poate fi: cea mai rapida cale sau cea mai scurta cale.

Similar functioneaza si un protocol de rutare (OSPF, RIP etc.), care determina cea mai scurta cale catre o anumita destinatie (ex: Google.ro). El isi poate alege pe post de **metrica** cea mai scurta cale (pe baza numarului de Routere sau hopuri), cea mai rapida cale (dpvd. al vitezei legaturilor).

Next-Hop si interfata de iesire - rutare recursiva

Daca ar fi sa simplificam mult lucrurile, Routerul trebuie sa faca un singur lucru: **sa primeasca un pachet pe o interfata** (sa-l proceseze) si **sa-l trimita mai departe pe o alta interfata**.

Acesta face legatura cu alte retele (astfel formand Internetul). Pentru a trimite un pachet de la o sursa la o destinatie, Routerul trebuie sa stie **CUI** trebuie sa-i trimita (sau mai exact pe ce interfata). Sa presupunem ca avem urmatoarea topologie de retea:

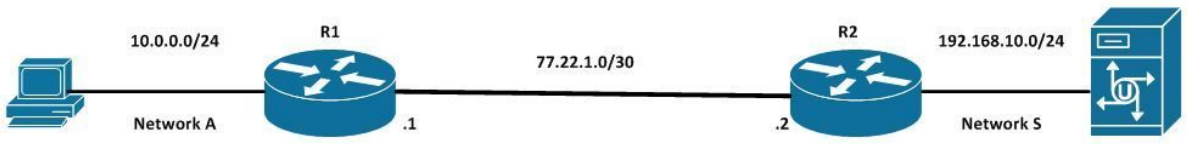


Figura 9.4

Noi vrem sa trimitem un mesaj de la PC-ul din reteaua A (IP: 10.0.0.9) la serverul din reteaua S (IP: 192.168.10.11).

In momentul in care mesajul ajunge la R1, acesta trebuie sa ia o decizie (mai exact trebuie sa decida catre cine va trimite acel mesaj astfel incat acesta sa ajunga la destinatie (Reteaua Serverului)).

R1 se uita in tabela lui de rutare si se intreaba daca are o ruta (cale) catre reteaua 192.168.10.0/24. Daca exista aceasta ruta, R1 va trimite mai departe mesajul (catre R2). Altfel daca ruta nu exista, mesajul va fi aruncat.

In momentul in care mesajul ajunge la R2, acesta il va trimite mai departe (spre destinatie) catre Server.

Capitolul 10 - Rute Statice

Pe tot parcusul acestui capitol vom discuta despre Rute Statice, mai exact: **De ce** avem nevoie de ele, care sunt **avantajele** si **dezavantajele** acestora si cum se **configureaza**. Pentru inceput vom folosi topologia de mai jos pentru a intelege conceptul:

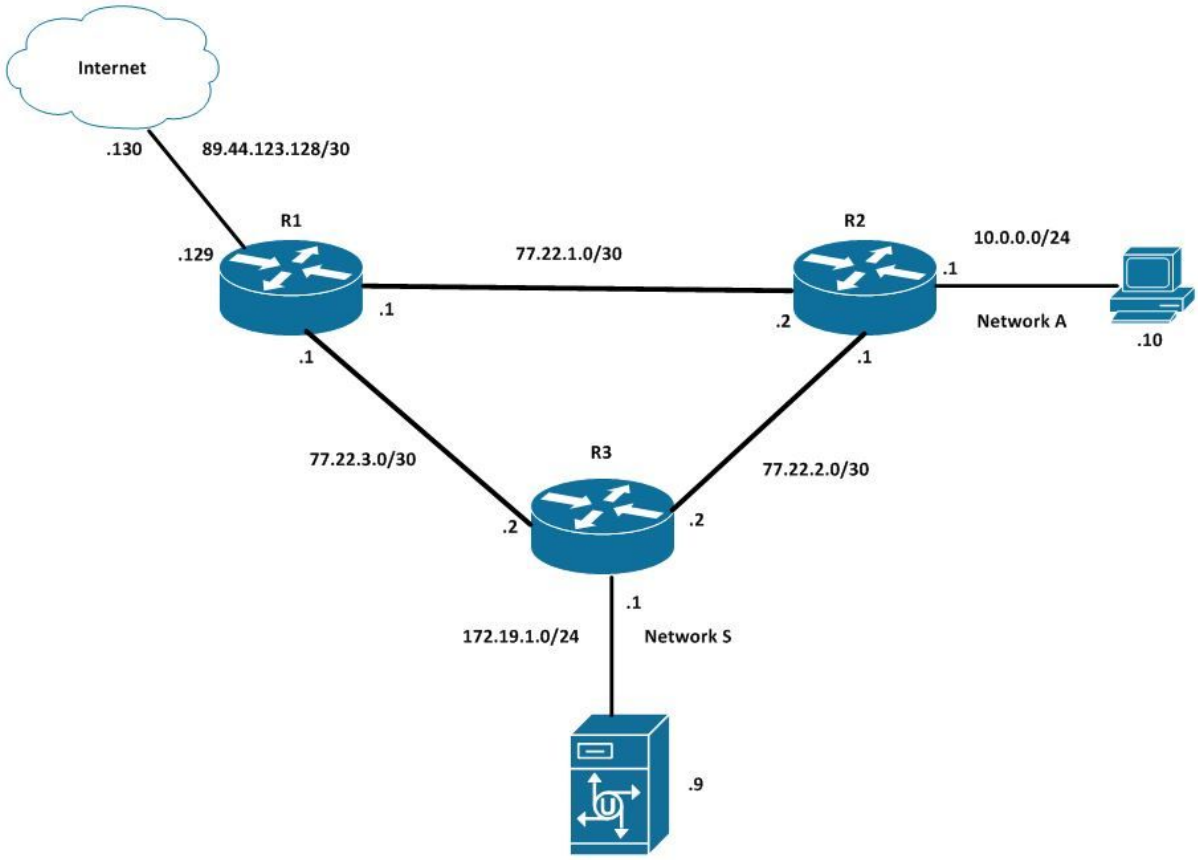


Figura 10.1

De ce avem nevoie de Rute Statice ?

By default, un Router **cunoaste** doar *retelele Direct Conectate*. Acesta nu stie cum sa trimita mai departe de aceste retele, pachetele. Aici intervenim noi, cei care administram aceste echipamente si configuram rutele pe device.

In momentul in care **porneste**, un Router, invata mai intai de retelele direct conectate (cele care incep cu C, in tabelul de mai jos).

In figura de mai jos poti vedea Tabela de Rutare a unui Router Cisco, care contine rutele/retelele direct conectate (C) si adresa IP a lui R2 de pe acele interfete (L).

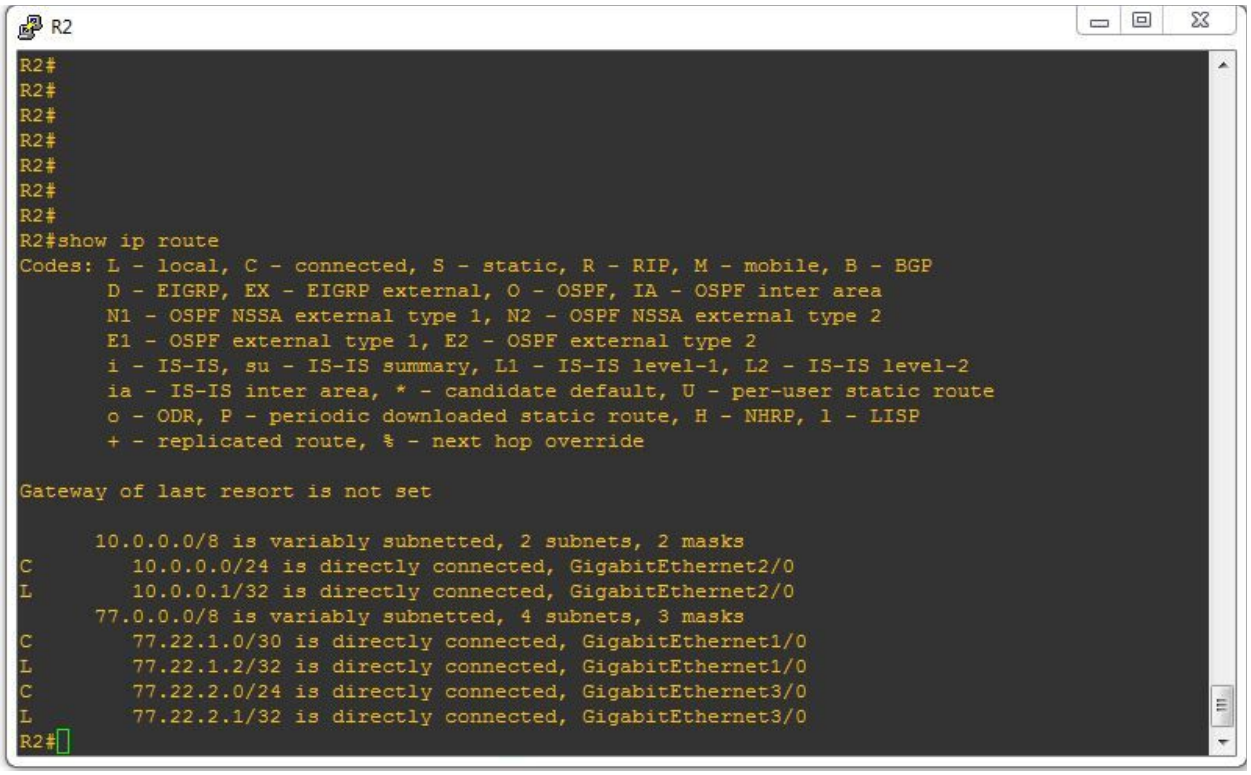


Figura 10.2

Dupa cum spunea, Routerul nu stie cum sa trimita (mai departe de retelele direct conectate) traficul. Aici intervenim **noi**, cei care administram aceste echipamente si configuram rutele pe device. Exista **2 moduri** prin care ii putem spune unui Router cum poate ajunge la o anumita retea (destinatie):

1) **Manual** - folosind Rute Statice

2) **Dinamic** - Routerele comunica intre ele si stabilesc cea mai buna (rapida) cale catre o retea (destinatie)

Rutele Statice sunt foarte importante cand vine vorba de *conectivitatea* intre retele (sau in Internet). In momentul in care introducem o ruta statica in tabela de rutare, aceasta va aparea in felul urmator:

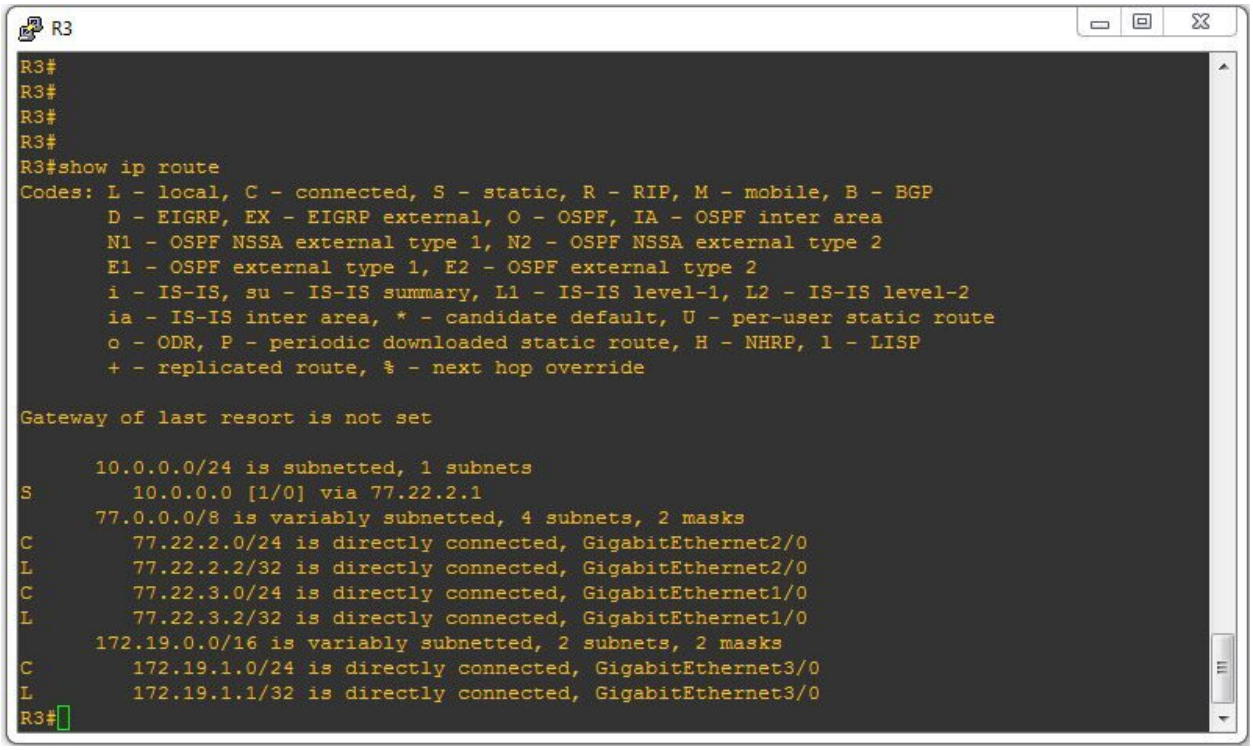


Figura 10.3

In tabela de rutare din figura de mai sus, ruta statica este: **S 10.0.0.0 [1/0] via 77.22.2.1** lata ce reprezinta acestea:

- *S* reprezinta tipul rutei (si anume **Static**)
- *10.0.0.0/24* reprezinta reteaua destinatie
- [*1/0*] - **1** reprezinta **AD** (Administrative Distance), iar **0** reprezinta **Metrica**

Avantaje / Dezavantaje

Exista 2 moduri prin care un Router poate invata despre o anumita retea:

- Ruta Statica
- Protocol Dinamic de Rutare (vom vedea in Capitolul 3)

Avantajele Rutelor Statice fata de Protocoalele Dinamice

- **Setate manual** de catre un *administrator* (astfel adaugand un nivel in plus de securitate)
- Sunt **potrivite pentru retele mici** (more networks = more static routes).
- **Nu consuma resurse** (CPU, RAM)

Dezavantajele Rutelor Statice fata de Protocoalele Dinamice

- **Nu** se adapteaza automat la schimbarile din retea (necesita interventia administratorului)
- **Setarea** devine **complexa** odata cu cresterea retelei
- Pot aparea foarte usor **erori de configurare**

Cum configuram Rutele Statice ?

In aceasta parte vom vedea cum putem *configura rutele statice* (urmand sa-ti perfectionezi cunostintele cu ajutorul celor 2 laboratoare de la final). Sa luam urmatorul scenariu:

Sa presupunem ca PC-ul din reteaua A vrea sa accese serviciile (Web, Transfer de Fisiere, etc.) de pe serverul din reteaua S, iar noi trebuie sa facem acest lucru posibil.

Dupa cum spuneam si la inceputul acestui capitol, Routerele, by default, nu stiu de exista altor retele in afara celor direct conectate (in **cazul lui R2** – *reteaua A, reteaua dintre R2* – R3, iar in **cazul lui R3** - *reteaua S, reteaua dintre R2 si R3*). Astfel, R2 stie cum sa ajunga la R3 dar **nu stie cum sa ajunga** in reteaua S.

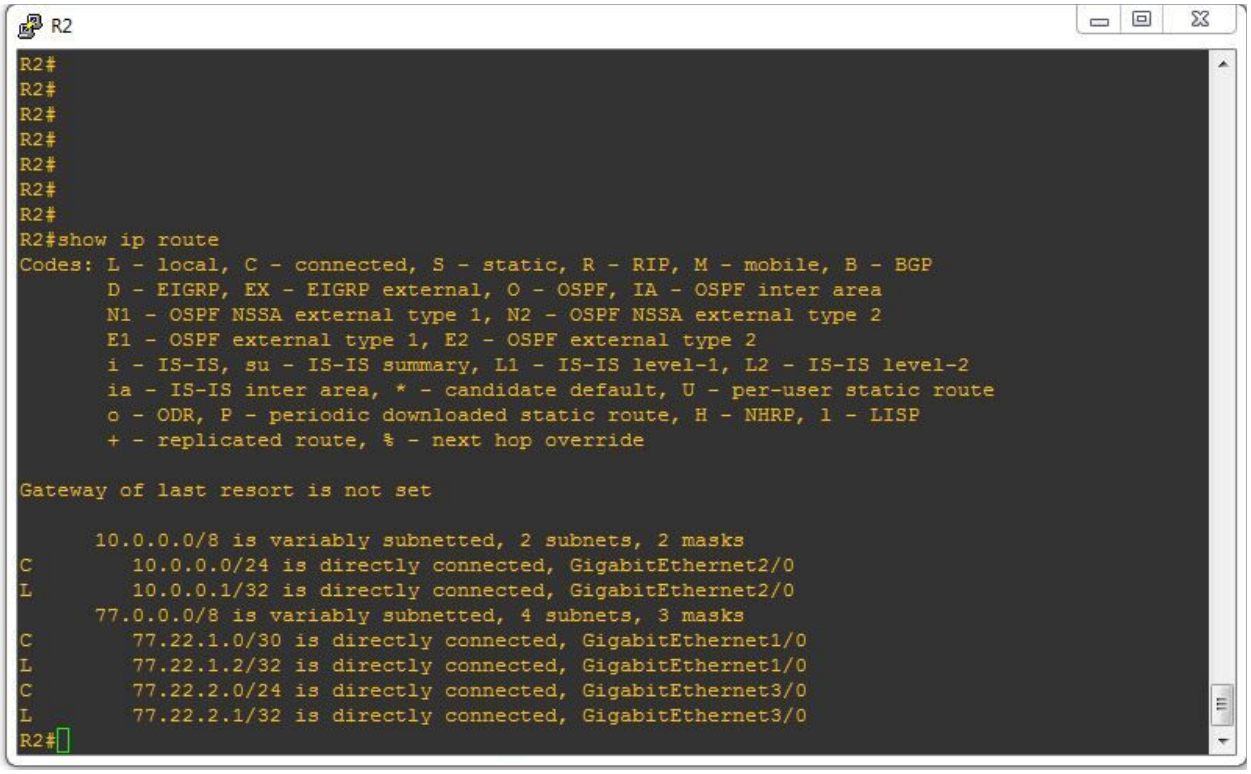


Figura 10.4

Noi vom seta o ***ruta statica*** pe R2, catre reteaua S (172.19.1.0/24). Acest lucru **nu este de ajuns** pentru a trimite trafic intre cele 2 retele deoarece:

1. R2 stie cum sa trimita trafic catre server
2. **Dar** traficul de return (intoarcere de la Server catre PC) se va bloca la R3 pentru ca, acesta, nu stie cum sa ajunga in reteaua A (10.0.0.0/24).

Traficul initial: **A->R2->R3->S**

Traficul de intoarcere: **S->R3->?** (*Rezultatul* => **Drop**)

Pentru **a seta o ruta statica** pe R2 catre reteaua S (cea a serverului), respectiv pe R3 catre reteua A vom introduce urmatoarele comenzi:

```
R2(config)#ip route 172.19.1.0 255.255.255.0 77.22.2.2
```

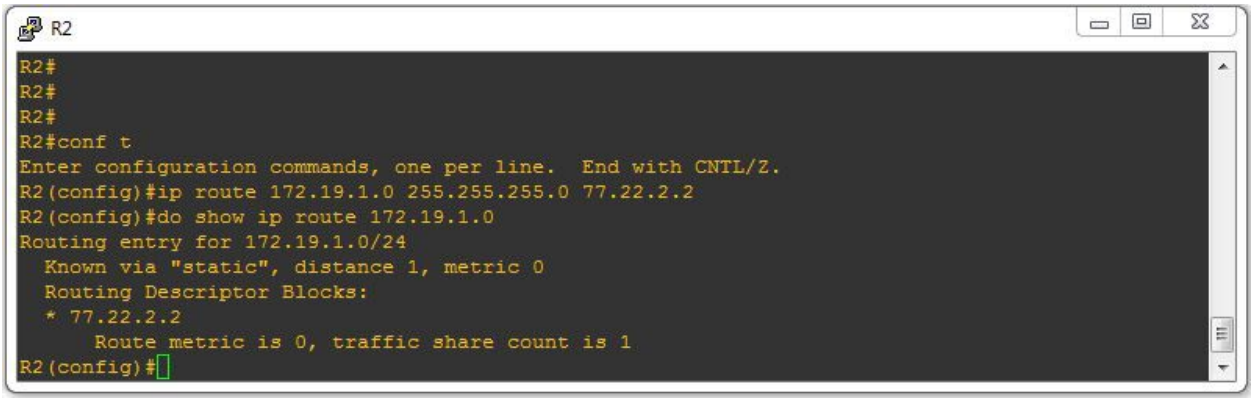


Figura 10.5

```
R3(config)#ip route 10.0.0.0 255.255.255.0 77.22.2.1
```

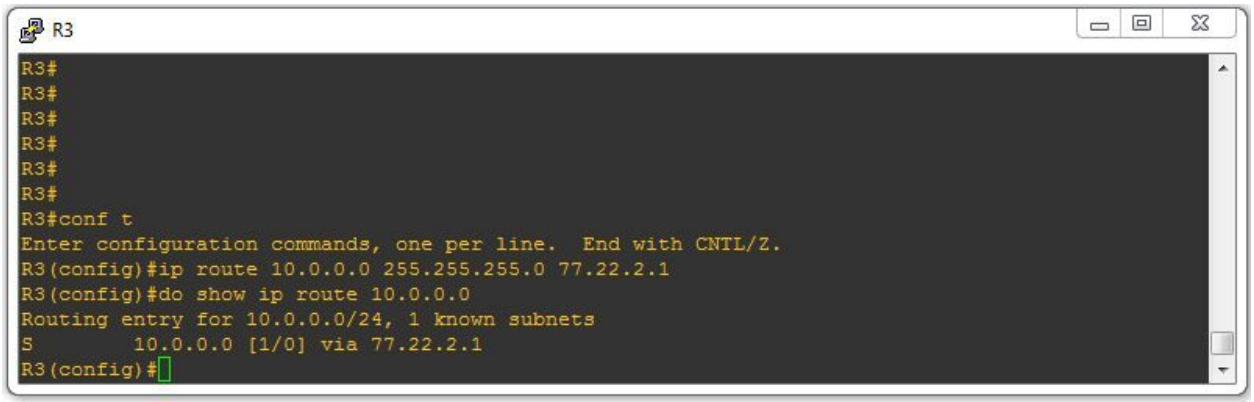


Figura 10.6

Structura acestei comenzi arata in felul urmator:

```
R1(config)#ip route DESTINATION_NETWORK   MASK   NEXT_HOP
```

DESTINATION_NETWORK = unde vreau sa ajung (pentru R2 va fi 172.19.1.0/24 (S), iar pentru R3 va fi 10.0.0.0/24 (A)

MASK = masca de retea a retelei destinatie (pentru /24 ea va fi 255.255.255.0)

NEXT_HOP = specific prin cine ajung la reteaua destinatie. R2 va ajunge la reteaua S prin R3

Ruta Default (0.0.0.0/0)

Exista o *ruta statica speciala*, aceasta se numeste **Default Route** (aka **0.0.0.0/0**). Rolul ei este de a-l informa pe Router sa trimita tot traficul (spre o destinatie pe care nu o cunoaste). Aceasta **ruta este utila** in momentul in care avem o conexiune catre Internet printr-un singur Router si dorim sa trimitem tot traficul (destinat Internetului) catre acest Router.

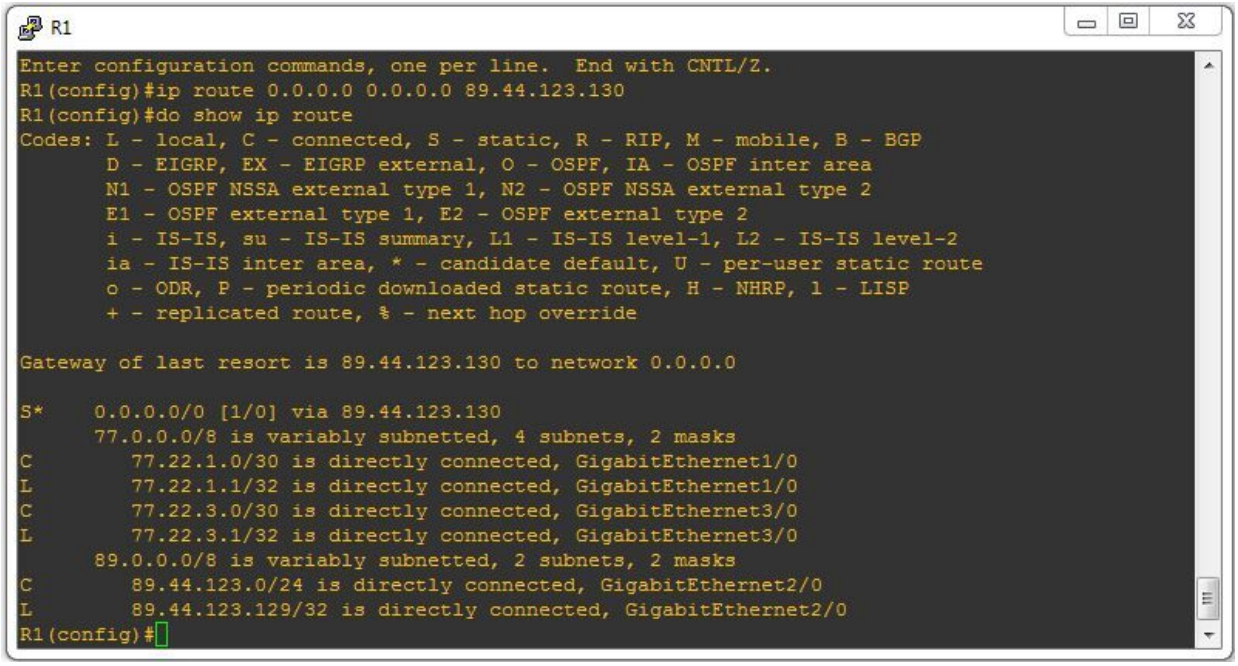


Figura 10.7

In exemplul de mai sus, R1 este conectat la Internet, iar noi am setat o ruta statica default (cea cu **S***):

```
R1(config)#ip route 0.0.0.0 0.0.0.0 89.44.123.130
```

Astfel daca **dorim sa ajungem la google.ro** (sau la orice alta resursa din Internet) vom putea face asta ! Tot ce mai trebuie sa facem este **sa adaugam rute statice default** si pe **R2**, respectiv **R3** catre R1 astfel incat PC-ul si serverul sa poata accesa Internetul.

```
R2(config)#ip route 0.0.0.0 0.0.0.0 77.22.1.1
R3(config)#ip route 0.0.0.0 0.0.0.0 77.22.3.1
```

Verificarea Setariilor

Pentru a verifica daca totul merge asa cum trebuie avem mai multe comenzi la dispozitie cu ajutorul carora putem verifica/testa, daca ceea ce am configurat functioneaza. De pe R2 vom da urmatoarele comenzi:

```
R2#show ip route
```
//pentru a vedea tabela de rutare

```
R2#show run | section route
```
// pentru a verifica comenziile introduse mai devreme

```
R2#ping 89.44.123.129
```
// pentru a vedea daca avem access la reteaua cu Internet

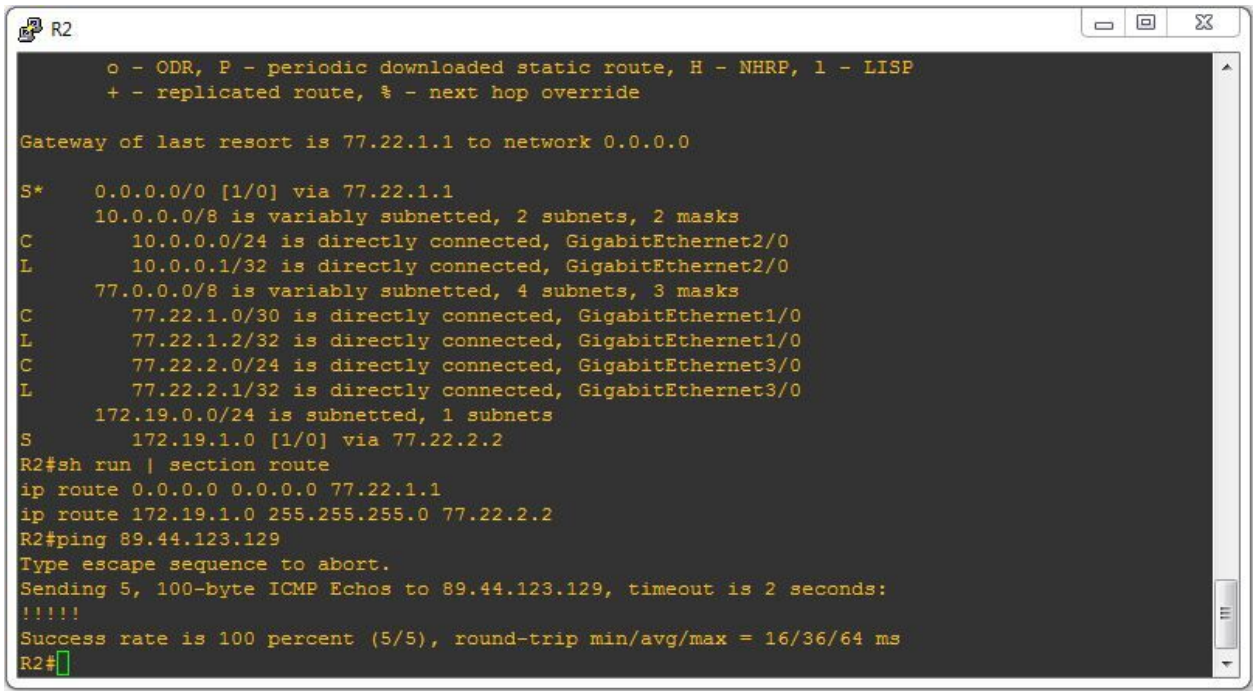
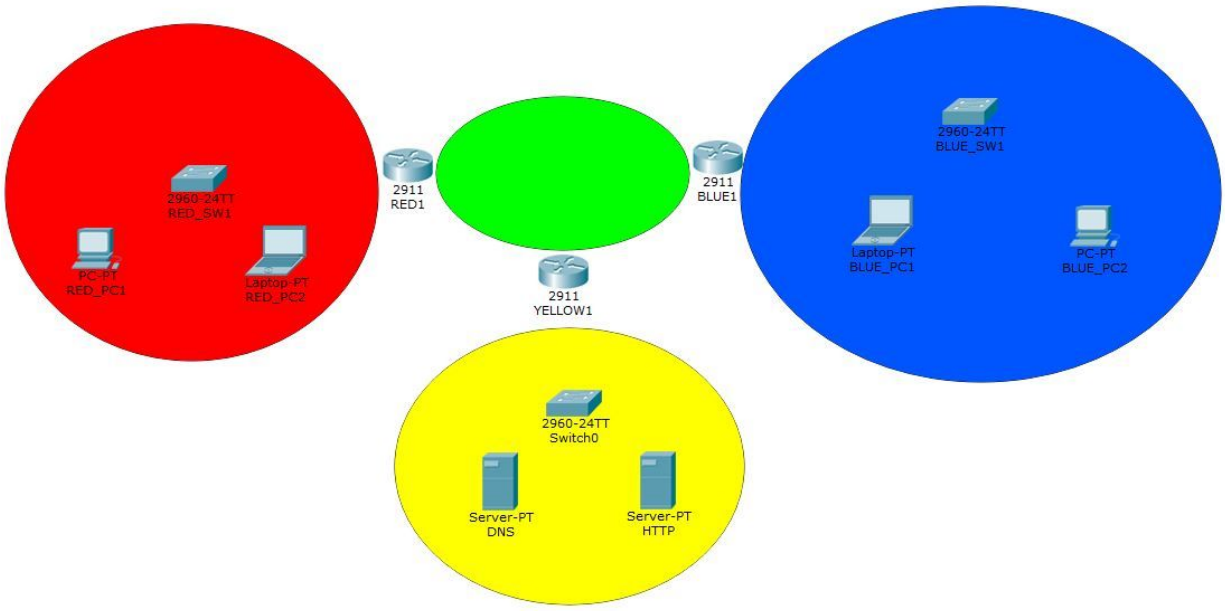


Figura 10.8

Laboratorul #2

Acum am ajuns la partea de laborator (partea practica), care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurari de baza ale Rutelor Statice pe Routere



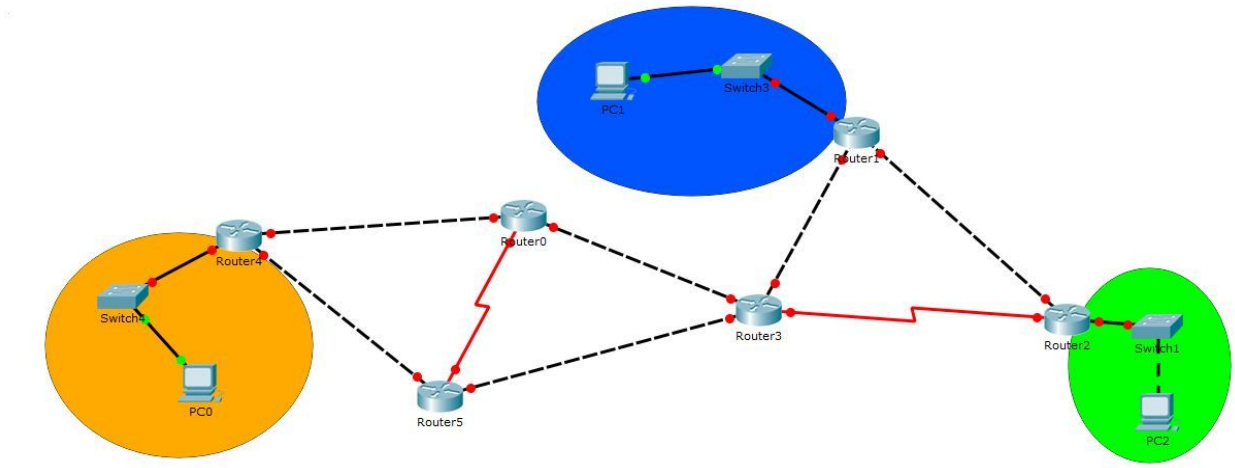
SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Laboratorul #3

Acum urmeaza sa aprofundam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aprofundarea cunostintelor de Rutare Statica pe o retea mai mare



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Capitolul 11 - Protocoale de Rutare

Distance Vector vs Link State

Cand vorbim de protocoale dinamice de rutare, in general, ne referim la 2 clase:

- ***Distance Vector*** (RIP, EIGRP)
- ***Link State*** (OSPF, IS-IS)

Distance Vector sunt acele protocoale de rutare care se bazeaza pe principiul “Routing by rumour” (adica ma bazez pe **informatia primita de la vecin**, nu am o vedere de ansamblu a intregii retele). Din aceasta categorie fac parte protocoale precum *RIP si EIGRP* (acesta este considerat mai de graba un *protocol hibrid*).

Link State sunt protocoalele de rutare care au o vedere de ansamblu (o harta activa) a intregii retele. Stiu “tot ce misca”. Au informatii despre fiecare link in parte (starea lui curenta: up/down; viteza acelui link – 100Mbps, 1Gbps etc.). Din aceasta categorie fac parte protocoalele *OSPF si IS-IS*.

Distance Vector

Ce este RIP (Routing Information Protocol) ?

Practic la inceput (dupa ce a terminat procesul de boot/pornire) fiecare Router cunoaste **DOAR** retelele direct conectate. Odata ce pornim RIP pe acestea, fiecare Router in parte ii va spune vecinului despre rutele sale direct conectate. Astfel fiecare Router va invata de cel putin 1 ruta a vecinului. In acest punct, echipamentele stocheaza toata aceasta informatie intr-o tabela - **Tabela de Rutare**.

Fiecare Router in parte, va trimite toate informatiile pe care le cunoaste despre retea (aka Tabela de Rutare) catre vecinii **sai direct conectati**. Procesul **se repeta** pana cand fiecare echipament care “ruleaza” RIP stie cum sa ajunga in fiecare punct al retelei.

Acesta proces de trimitere a tabelei de rutare (cunoscut ca **RIP Update**) este unul repetitiv si are loc la fiecare **30 de secunde** (Update Timer). Astfel intr-un interval de timp fiecare Router va sti de fiecare retea in parte.

Cum Functioneaza RIP ?

RIP stabilieste cea mai scurta cale pana la o retea destinatie pe baza metricii. In acest caz **metrica este numarul de hop-uri** (mai exact, prin cate Routere trebuie sa trec pentru a ajunge la destinatie ?)

RIP are si anumite dezavantaje cum ar fi:

- O retea poate fi la **Maxim 15** hop-uri distanta
- **Sumarizeaza automat** retelele (exista riscul care Routerele sa nu invete toate retelele)
- Update-uri rare (la **30 de secunde**)
    - O cadere (downtime) a retelei de 20 de secunde poate implica pagube financiare

Sa luam exemplul de mai jos (folosit si mai devreme):

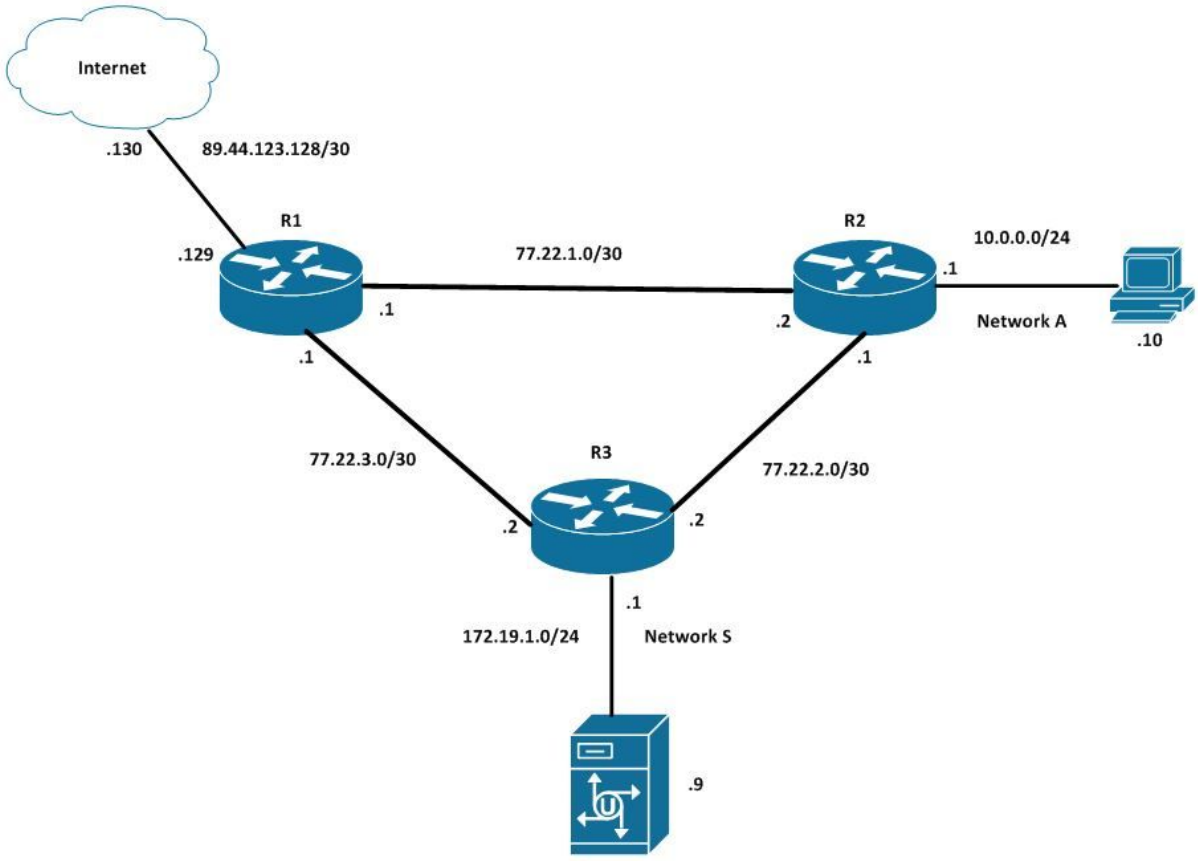


Figura 11.1

Daca dorim sa trimitem trafic de la PC la Server, Routerul are 2 posibilitati:

1) Trimite traficul lui R3, care trimite Server-ului
    a) ***PC -> R2 -> R3 -> Server*** **(2 Hop-uri)**
2) Trimite traficul lui R1, care trimite mai departe lui R3, care trimite Server-ului
    a) ***PC -> R2 -> R1 -> R3 -> Server*** **(3 Hop-uri)**

Calea cea mai scurta, in acest caz, este prima. Asadar, R2 va instala in Tabela lui de Rutare:

172.19.1.0/24 [120/2] via 77.22.2.2 (IP-ul lui R3), unde *120 reprezinta AD-ul*, iar *2 reprezinta metrica*.

RIP are 2 versiuni, v1 care nu mai este folosit si **versiunea actuala v2** care este cea folosita (doar atunci cand este cazul, adica atunci cand se foloseste RIP). Dupa cum spuneam si mai devreme, RIP este un **protocol de rutare vechi** si destul de redus din

punctul de vedere al caracteristicilor, acesta fiind motivul pentru care nu mai este folosit (decat in anumite cazuri).

In diferite carti sau cursuri de retelistica, **scopul RIP-ului** este unul **didactic** pentru ca ilustreaza foarte bine modul de functionare al unui protocol de rutare (astfel trecerea de la RIP la OSPF sau EIGRP este mai usoara).

Configurare RIP pe Routere Cisco

Acum, sa vedem cum putem configura RIP pe un Router Cisco. Iata comenzile:
Pe R2:

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#network 77.22.2.0
R2(config-router)#network 77.22.1.0
R2(config-router)#network 10.0.0.0
```

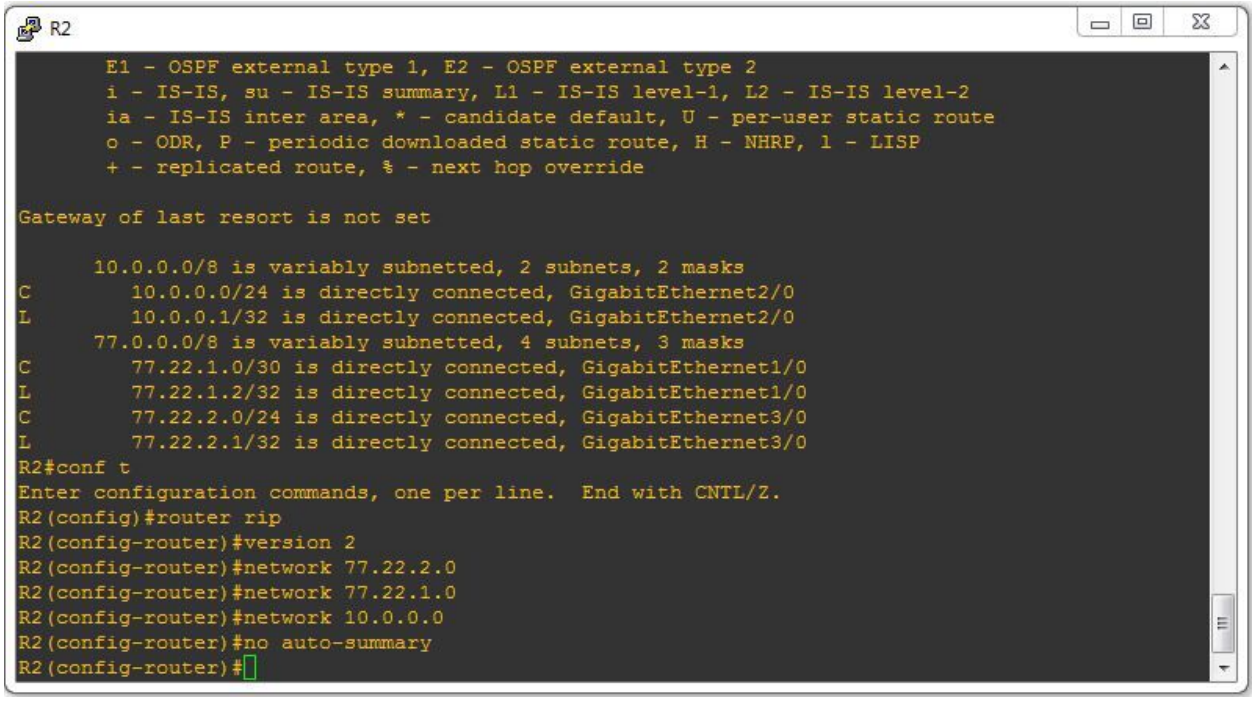


Figura 11.2

Pe R3:

```
R3(config)#router rip
R3(config-router)#version 2
R3(config-router)#no auto-summary
R3(config-router)#network 77.22.2.0
R3(config-router)#network 77.22.3.0
R3(config-router)#network 172.19.1.0
```

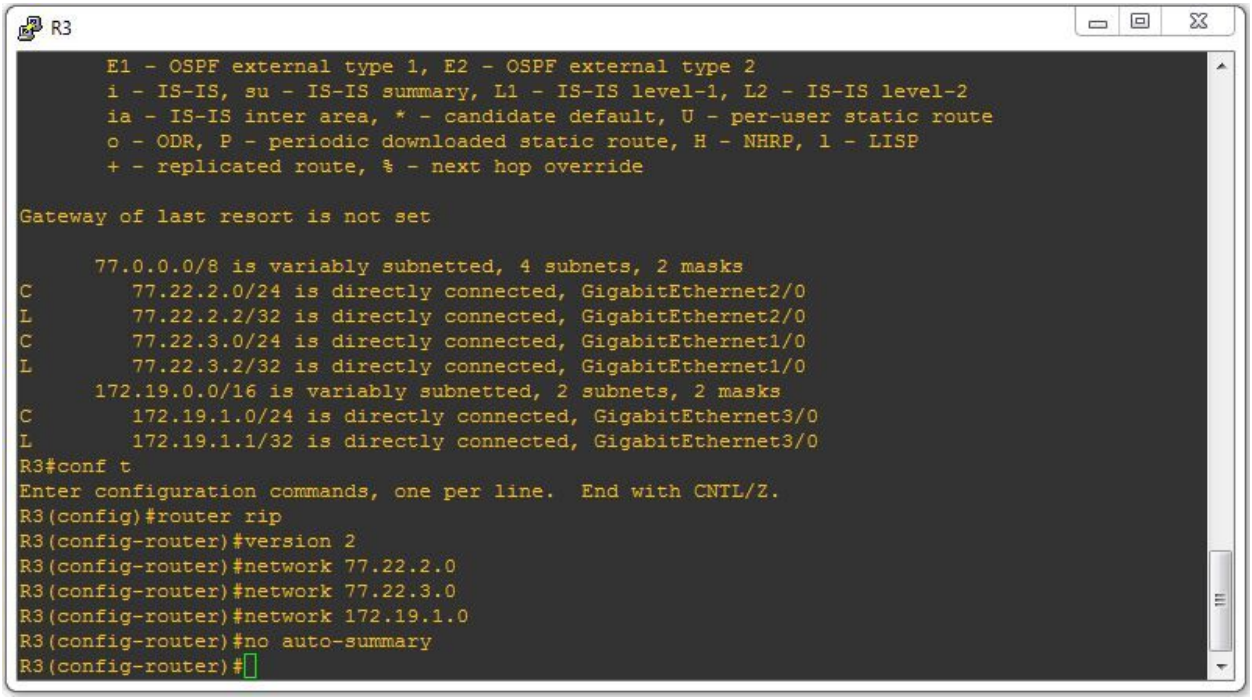


Figura 11.3

Pe R1:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 89.44.123.130
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 77.22.3.0
R1(config-router)#network 77.22.1.0
```

```
R1#
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 89.44.123.130
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#network 77.22.1.0
R1(config-router)#network 77.22.3.0
R1(config-router)#default-information originate
R1(config-router)#
```

Figura 11.4

Verificarea Configuratilor

Odata ce am facut setarile este important sa verificam (si sa intelegem) ce am facut pana acum. In figura de mai jos poti vedea, cum pe R2, am folosit comanda ***#show ip route*** pentru a verifica tabela de rutare (in care poti vedea **retelele invatate dinamic** scrise cu litera **R**).

R2

```
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 77.22.1.1 to network 0.0.0.0

R*    0.0.0.0/0 [120/1] via 77.22.1.1, 00:00:08, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.0.0.0/24 is directly connected, GigabitEthernet2/0
L        10.0.0.1/32 is directly connected, GigabitEthernet2/0
      77.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        77.22.1.0/30 is directly connected, GigabitEthernet1/0
L        77.22.1.2/32 is directly connected, GigabitEthernet1/0
C        77.22.2.0/24 is directly connected, GigabitEthernet3/0
L        77.22.2.1/32 is directly connected, GigabitEthernet3/0
R        77.22.3.0/24 [120/1] via 77.22.2.2, 00:00:08, GigabitEthernet3/0
R        77.22.3.0/30 [120/1] via 77.22.1.1, 00:00:08, GigabitEthernet1/0
      172.19.0.0/24 is subnetted, 1 subnets
R        172.19.1.0 [120/1] via 77.22.2.2, 00:00:08, GigabitEthernet3/0
R2#ping 172.19.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.19.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/34/52 ms
R2#
```

Figura 11.5

Urmatoarea comanda pe care am folosit-o este ***#ping 172.19.1.1***, cu ajutorul careia am *testat conectivitatea* catre **o ruta invatata dinamic**.

O alta comanda extrem de utila este ***#show ip protocols*** cu ajutorul careia putem afla mai multe informatii despre protocoalele de rutare functionale pe Router. In figura de mai jos (pagina urmatoare) poti veda:

- *Protocolul de rutare* este **RIP** si foloseste versiunea **2**
- *Rutarea este* ***pornita*** pe cele 3 interfete GigabitEthernet
- *Retelele sumarizate* pentru care a fost data comanda **network**
- *Distanta administrativa* (**AD**) a RIP-ului - **120**
- *Timerele protocolului* (*30s* - Update-uri, *180s* hold down timer)

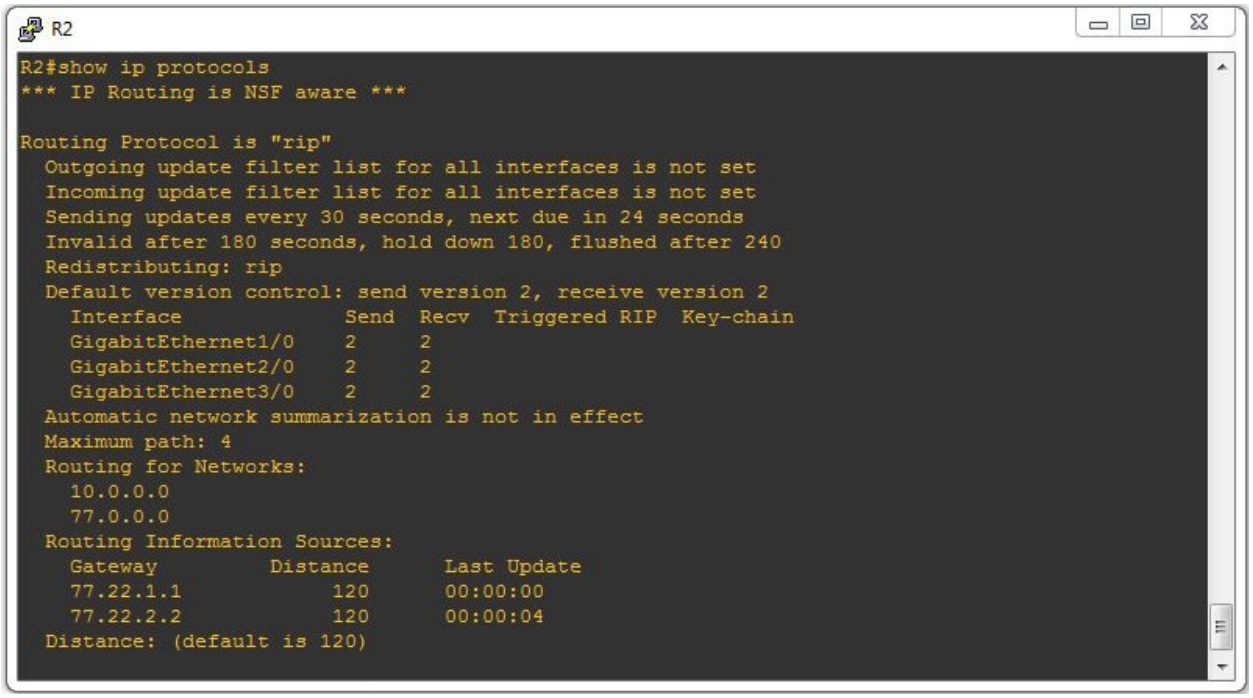


Figura 11.6

Asadar comenzile recomandate pentru verificarea setarilor sunt:

```
R2#show ip protocols
R2#show ip route
R2#show run | section rip
```

Procesul de Selectie a unei Rute din Tabela de Rutare

In momentul in care Routerele trebuie sa ia decizia de rutare, acestea cauta cea mai buna cale spre destinatie in tabela de rutare. Routerul va parcurge tabela de rutare pe baza unor algoritmi care incearca sa reduca timpul cautare. Dar la un moment dat se intampla asta:
IP Destinatie: 172.16.0.10

Iar Routerul are disponibil **3 retele diferite**:

*172.16.0.0/**16*** via GigabitEthernet **0/1**

*172.16.0.0/**18*** via GigabitEthernet **0/2**

*172.16.0.0/**26*** via GigabitEthernet **0/3**

Practic Routerul trebuie sa decida care dintre acestea va fi reteaua destinatie astfel incat sa stie pe ce interfata sa trimita pachetul. Routerul va face ceea ce se numeste *Longest Match Routing,* adica va lua reteaua cu **masca cea mai mare** (cu cei *mai multi biti de retea*) si o va alege pe aceea ca retea destinatie pentru ca o considera **reteaua mai specifica.** Hai sa luam exemplul de mai jos pentru a intelege mai bine:

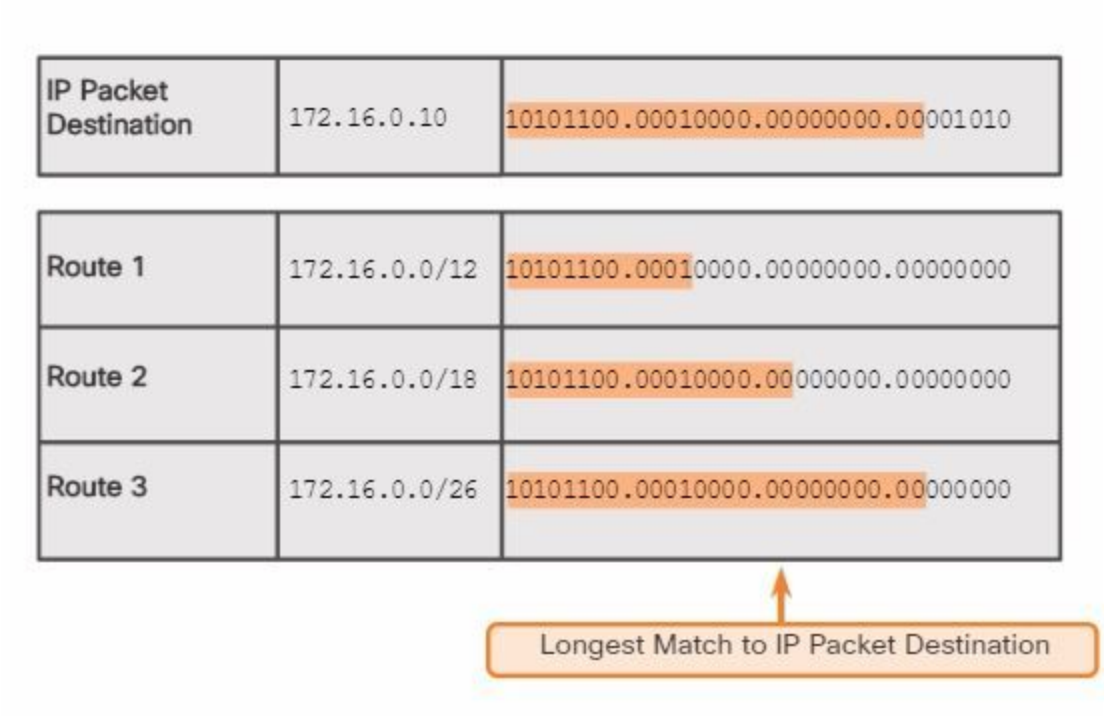


Figura 11.7

Astfel a 3-a ruta avand cei mai multi biti de retea in comun (masca cea mai mare) va fi aleasa calea spre destinatie pentru host-ul 172.16.0.10

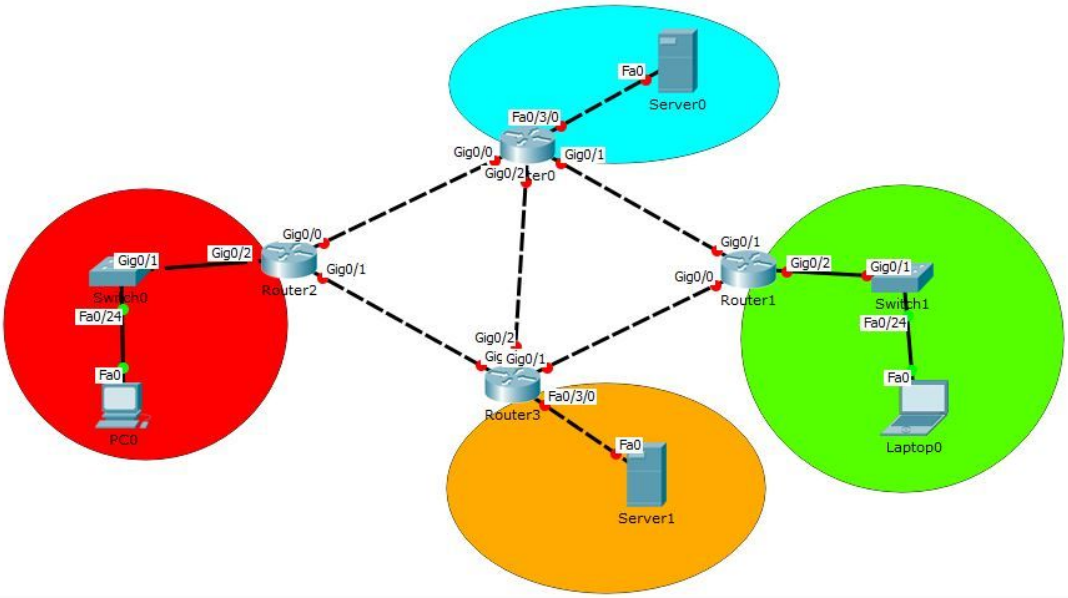
Concluzie

RIP este un protocol care **era folosit pentru retele mici** (maxim 15 hop-uri distanta). In ziua de astazi acest protocol nu mai este folosit (decat in anumite situatii), dar este foarte **util** pentru a intelege **cum functioneaza Rutarea Dinamica**. Tot odata am aflat si faptul ca Routerele (cand au o decizie de facut) aleg cea mai specifica masca atunci cand sunt mai multe posibilitati spre o destiantie.

Laboratorul #4

Acum am ajuns la partea de laborator (partea practica), care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurari de baza pentru RIP



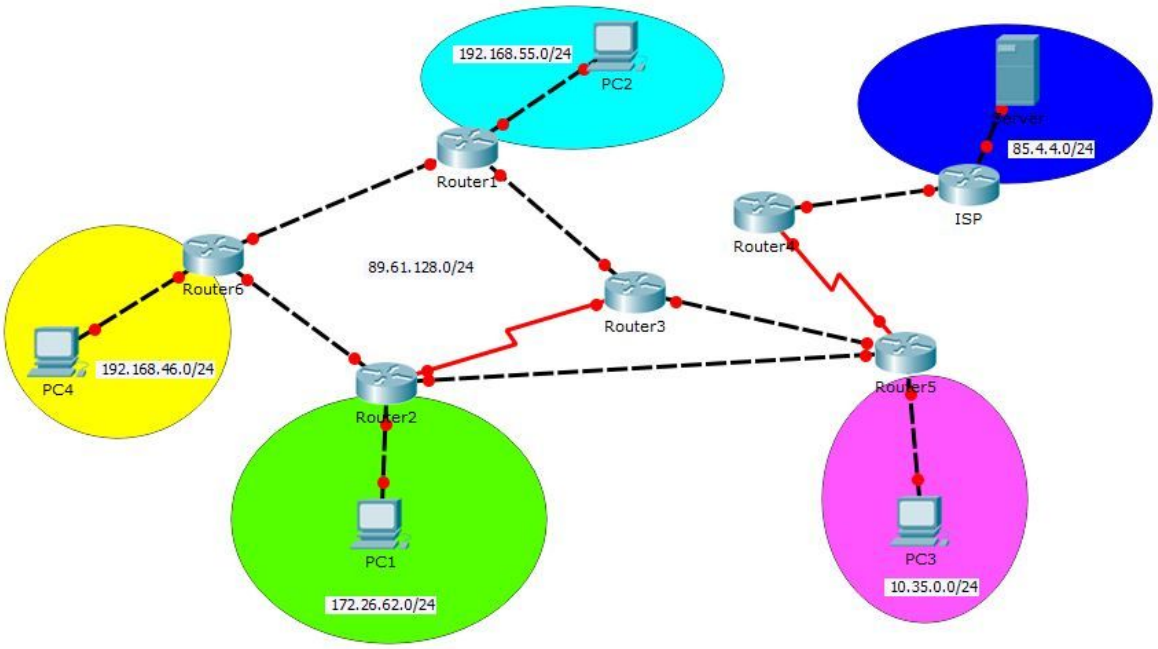
SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Laboratorul #5

Acum am ajuns la partea de laborator (partea practica), care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aprofundarea cunostintelor de protocoale de rutare (RIP)



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Capitolul 12 - Link-State

In cazul protocoalelor Link-State (OSPF, IS-IS), lucrurile stau putin diferit. Acestea au o **vedere de ansamblu** asupra intregii retelei. Ele cunosc informatii precum:

- **Starea** fiecarei retele (up/down) si a Routerelor
- **Viteza** acestora (100 Mbps, 1Gbps, 10Gbps, etc.)

Astfel, avand aceste informatii (pe care le stocheaza intr-o baza de date specifica fiecarui Router in parte), Routerele isi **construiesc o harta a retelei** si o **actualizeaza constant** pe masura ce apar modificari in retea.

OSPF (**O**pen **S**hortest **P**ath **F**irst)

OSPF este **cel mai folosit protocol de rutare** in retelele mari, medii si mici din Internet. Popularitatea acestui protocol este datorata faptului ca este **vendor-independent** (adica oricine il poate folosi/instala pe orice echipament). Aceste este deosebit de ***EIGRP ,*** pe care unii il considera mult mai bun si eficient, si care este **dezvoltat de Cisco**, putand fi folosit DOAR pe echipamentele Cisco.
OSPF face parte din **categoria IGP** (**I**nterior **G**ateway **P**rotocol), impreuna cu protocoale RIP, EIGRP si IS-IS. Singurul protocol care apartine celeilalte categorii (EGP - Exterior Gateway Protocol) este BGP, protocol folosit pentru rutarea Internetului. OSPF foloseste algoritmul lui **Dijkstra** pentru calcularea celor mai bune cai catre retelele destinatie. Prin schimbul de mesaje intre Routere, acestea reusesc sa-si faca o harta asupra intregii retele. Cunoscand fiecare retea din **intreaga topologie**, Routerele pot calcula cea mai scurta cale pana la o anumita destinatie pe baza costului.

Cum functioneaza OSPF ?

Metrica in OSPF

*“**Costul** unei retele depinde de viteza acelei retele.”*

Costul se calculeaza astfel:
Cost = 1000 / BW unde BW reprezinta viteza (10, 100, 1000 Mbps).

De exemplu: Sa determinam costul de la A la B, pentru o reteaua de mai jos:

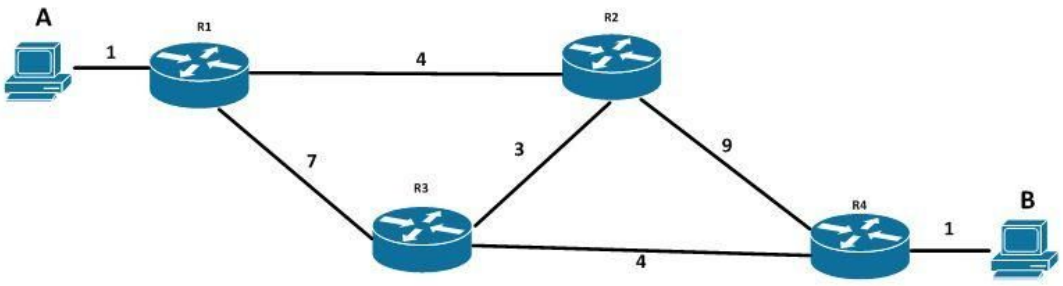


Figura 12.1

Sa presupunem ca R1 trebuie sa determine cea mai rapida cale (cu cel mai mic cost) spre reteaua destinatie B. Hai sa vedem care sunt variantele posibile pentru acesta:

- *R1 -> R2 -> R4 -> B: 14*
- *R1 -> R3 -> R4 -> B: 12*
- *R1 -> R2 -> R3 -> R4 -> B: 12*
- *R1 -> R3 -> R2 -> R4 -> B: 20*

Asadar, iata ca avem 4 cai posibile, dar R1 trebuie sa o aleaga pe cea mai buna. Cea mai buna cale va fi considerata cea cu costul cel mai mic pana la destinatie, adica:

R1 -> R3 -> R4 -> B: 12 si **R1 -> R2 -> R3 -> R4 -> B: 12**

Iar acum intrebarea fireasca ar fi: “Poate R1 sa aiba 2 drumuri diferite pana la destinatie ?” Raspunsul este da, clar, acest lucru se poate intampla. Comportamentul lui R1 va fi acela de a face ***load-balancing*** (va balansa traficul total, o conexiune va trimite catre R2, iar cealalta catre R3). In **concluzie**, fiecare retea are un anumit cost (100, 10 , 1 - depinde de viteza), iar costul total de la o retea (sursa) A pana la o retea (destinatie) B va fi suma costurilor (costurile acumulate). **Metrica folosita de OSPF este costul** (mai exact *suma costurilor de la sursa A la destinatia B*).

Distanta Administrativa a OSPF-ului

Pe echipamentele Cisco, **OSPF are AD-ul cu valoarea 110**. Asta inseamna ca, in momentul in care un Router primeste aceeasi informatie (ex: reteaua 10.87.33.0/24) din 2 surse diferite (ex: *RIP* (AD 120) si *OSFP* (110)), acesta va alege informatia venita din OSPF deoarece **protocolul sursa cu cel mai mic AD castiga** (110 < 120 => ruta din OSPF (110) va fi invatata si adaugata in tabela de rutare).

Tipurile de Mesaje din OSPF

Pentru a putea face schimbul de informatii de rutare, OSPF foloseste urmatoarele tipuri de mesaje:

- **Hello**
- **LSDB**
- **LSR, LSU & LSA**

O buna parte dintre aceste mesaje sunt trimise sub forma de **multicast** (de la o singura sursa catre un grup specific de Routere). Adrese IP destinatie (pentru multicast), folosite de catre OSPF sunt:

- ***224.0.0.6*** - folosit de Routerele DR si BDR (despre care vorbim mai tarziu)
- ***224.0.0.5*** - folosit de orice alte Routere fata de DR sau BDR

1) Mesajele de Tip Hello

Protocoalele Link-State (printre care si OSPF) au ceea ce se numeste o **tabela de adiacenta**. Aceasta tabela de adiacenta stocheaza informatia despre Routerele vecine direct conectate.

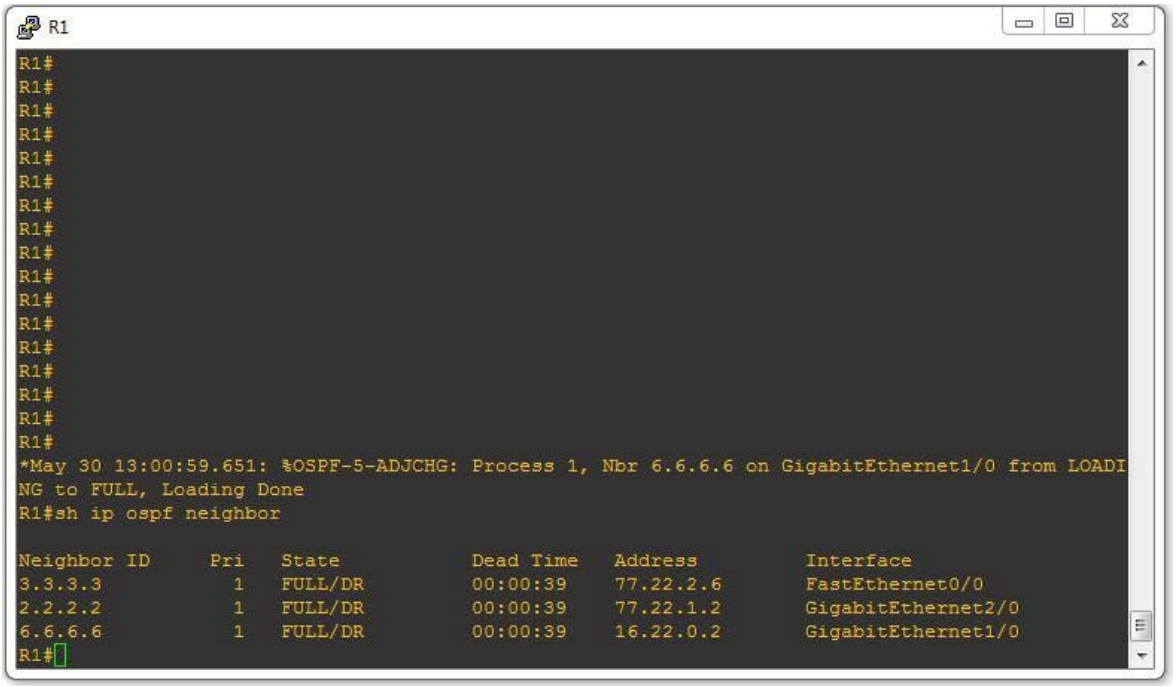


Figura 12.2

Inainte de a putea schimba informatii de rutare (retele), Routerele trebuie **sa stabilieasca o relatie de adiacenta** (mai exact trebuie sa cada de acord cu anumiti parametrii: *acceasi adresa de retea/masca, aceeasi arie, acelasi interval de timp* in care se trimit pachetele perioadice etc.).

*"**Relatiile de adiacenta** se stabiliesc intre Routere prin **mesaje speciale** numite **Hello**."*

Aceste Hello-uri se **trimit periodic** (odata la 10 secunde), in modul **multicast** (adica pentru un grup specific de dispozitive, mai exact cele care ruleaza activ procesul OSPF). In figura de mai jos (captura a traficului de pe Routere cu programul Wireshark) poti vedea cum arata un schimb de Hello-uri OSPF intre Routere:

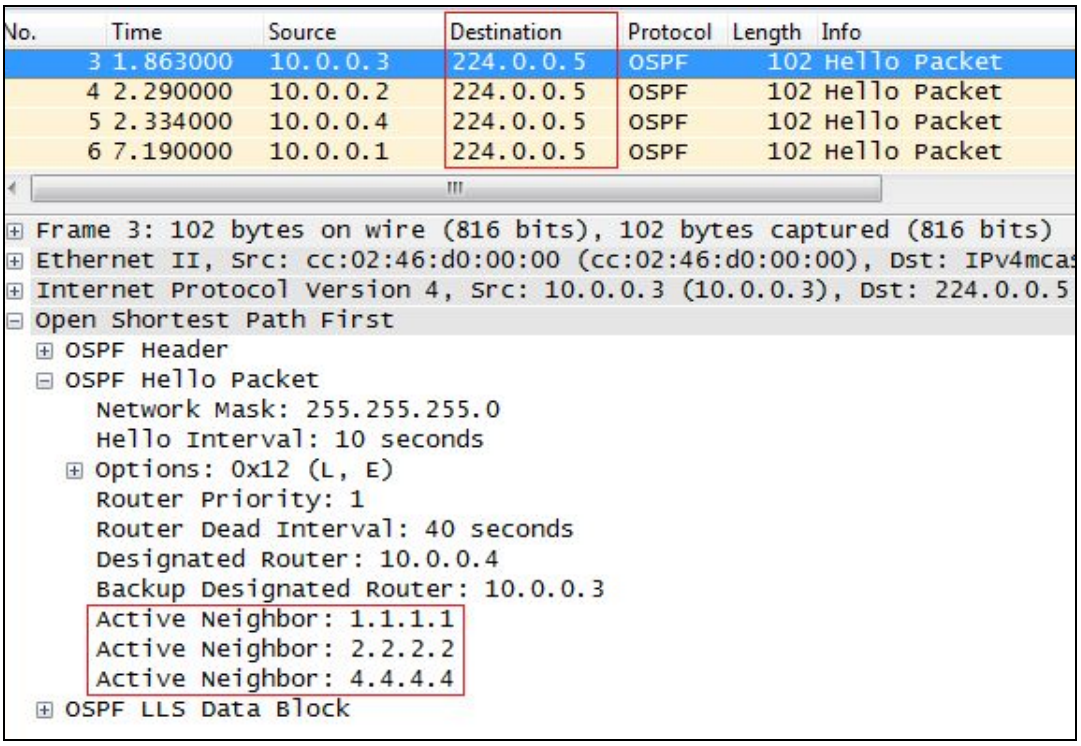


Figura 12.3

Pachetele sunt trimise pe ***adresa destinatie 224.0.0.5*** (adresa speciala multicast). Daca Routerele primesc pachetele Hello unul de la celalalt (R1 de la R2 si invers) si cad de acord cu parametrii aflati in acele pachete (parametrii pe care ii poti vedea in figura de mai sus) atunci R1 si R2 vor forma o relatie de adiacenta (si vor putea invata retele unul de la celalalt).
Daca au fost trimise 4 Hello-uri consecutive (adica 40 de secunde) si nu a fost primit un raspuns, aceasta relatie de adiacenta va fi intrerupta (si se vor pierde toate retelele invatate intre cele 2 Routere).

Timerele arata astfel:

- **10 secunde** - **Hello** Timer
- **40 secunde** - **Dead** Timer - se intrerupe relatia de adiacenta

2) Mesajele de Tip LSDB

Fiecare Router din OSPF are o viziune proprie asupra intregii retele. Asta inseamna ca ele pot lua decizii independent de celelalte routere. Pentru a putea face asta posibila, este necesara o **Baza de Date** (unde vor fi stocate toate aceste informatii). Asadar toata informatia de rutare va fi stocata intr-o **baza de date**.

Scopul OSPF-ului, pana la urma, este de **a face schimb** *de aceasta informatie* intr-un *mod cat mai eficient*. Astfel a fost conceput tipul de mesaj **LSDB** (Link-State Database) care pur si simplu trimite informatia (cu fiecare ruta in parte) catre vecinul direct conectat, lasand la “mana lui” sa aleaga de ce rute are nevoie.

Aceste LSDB-uri contin un **rezumat a bazei de date** (numite **DBD** - **D**ata**b**ase **D**escription). Avand acest proces, 2 Routere in OSPF pot cere unul de la celalalt doar retelele de care au nevoie (mai exact cele de care nu stiu inca).

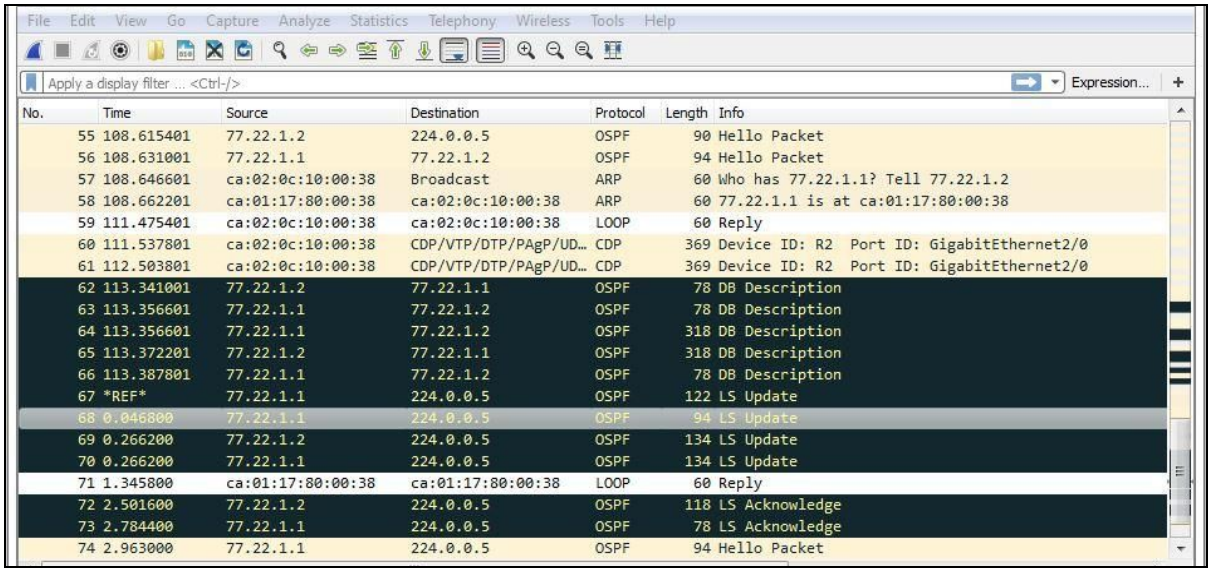


Figura 12.4

3) Mesajele de Tip LSR, LSU & LSA, LSAck

Dupa ce fiecare Router in parte a stabilit (prin schimbul de LSDB-uri) informatile de care are nevoie, este timpul ca acestea sa ceara, respectiv sa trimita informatiile de rutare.
LSR (*Link-State Request*) este tipul de pachete (folosit in OSPF) pentru a cere rutele de la un anumit vecin. Acest tip de mesaj este trimis unicast (se adreseaza direct unui Router vecin).
LSU (*Link-State Update*) este tipul de mesaj care contine retelele cerute de mesajul precedent (LSR). Acest mesaj este livrat Routerului vecin, tot in mod unicast. Un LSU poate contine *1 sau mai multe* **LSA**-uri (*Link-State Advertisement*). Aceste LSA-uri sunt de mai multe tipuri si contin fiecare un anumit tip de informatie de rutare.

LSA Type 1 - Router LSA

- contine informatie despre retelele din arie, stare si viteza lor

LSA Type 2 - Network LSA

- contine informatia trimisa de DR/BDR

LSA Type 3 - Summary LSA

- contine rutele din alte arii

LSA Type 4 - ASBR-Summary LSA

- contine cea mai scurta cale catre ASBR

LSA Type 5 - External LSA

- contine rutele care provin din afara OSPF-ului (ex: redistribuire din EIGRP, RIP etc.)

LSA Type 7 - NSSA External

- contine rutele care provin din afara OSPF-ului (ex: redistribuire din EIGRP, RIP etc.)

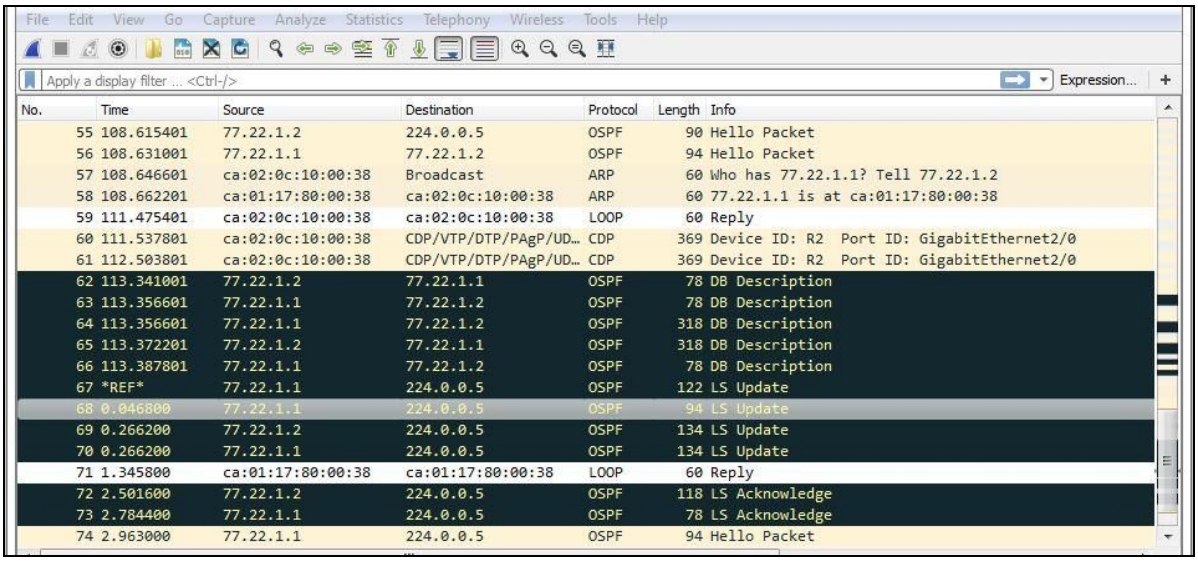


Figura 12.5

In figura de mai sus poti vedea captura in Wireshark a mesajelor OSPF enuntate mai devreme (mesaje precum LSU, DBD, LSAck). Astfel de mesaje sunt trimise in momentul in care 2 Routere formeaza o adiacenta (pentru ca acestea trebuie sa faca schimb de informatii) sau in momentul in care o ruta apare sau dispare din retea.

DR/BDR in OSPF

O problema clasica in OSPF (intr-o retea Ethernet) este cea cu transmiterea pachetelor OSPF. In cazul topologiei de mai jos (retea Ethernet), in momentul in care *Routerele trebuie sa faca schimb de informatii* acestea vor creea un **val** (dezorganizat) **de trafic** in retea, care poate **ingreuna** pe moment conectivitatea (atat *banda cat si CPU-ul Routerelor*).

Astfel, cei care au conceput OSPF-ul s-au gandit la o solutie pentru aceasta problema. S-au gandit la un mod mult mai organizat de transmitere a datelor si anume principiul de DR si BDR. Mai exact, un punct central in retea care va transmite mesajul de la fiecare Router in partea catre restul Routerelor din retea.

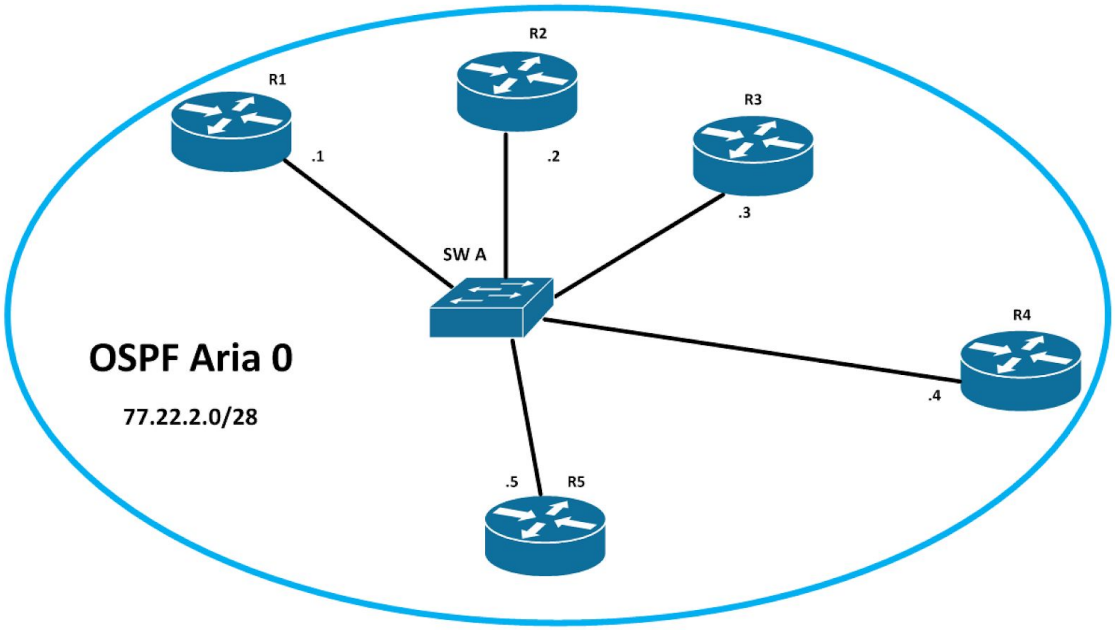


Figura 12.6

Acum, propun sa vedem la ce se refera conceptul de DR/BDR. In retea va exista un Router special, numit **DR** (*Designated Router*) si care va trimite mesajele pentru fiecare Router in parte (care participa in OSPF). Exista si un backup, numit **BDR** (*Backup Designated Router*), in cazul in care se intampla ceva cu DR-ul si dispare din topologie.

Scopul acestui design este de a trimite mesajele OSPF intr-un mod mai eficient si organizat cu un consum de latime de banda (si CPU) minim. Iata cum ar avea loc transmiterea mesajelor OSPF in reteaua de mai sus:

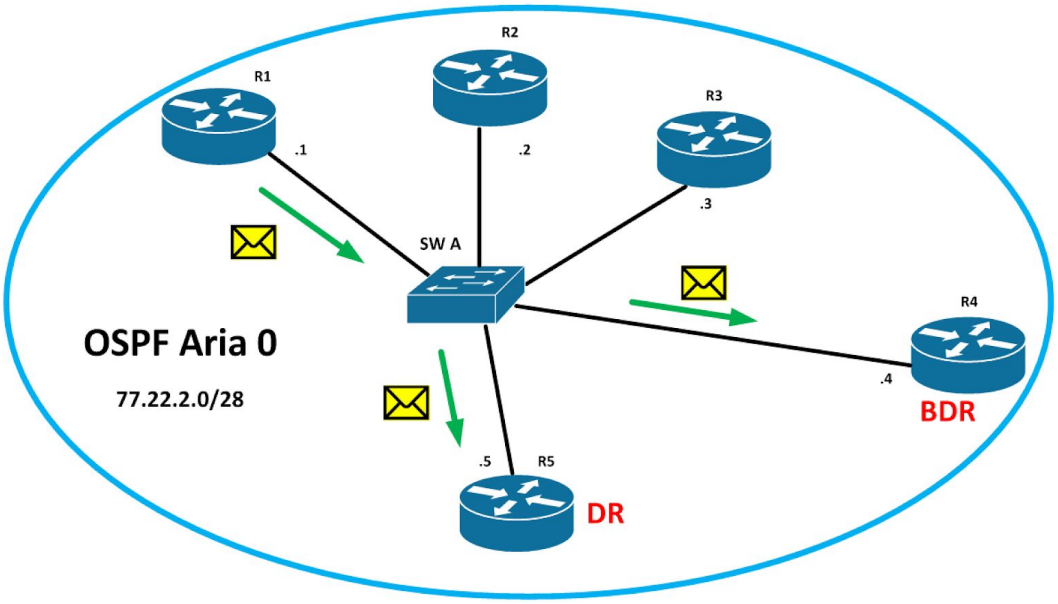


Figura 12.7

In figura de mai sus, R1 transmite un mesaj OSPF lui R5, respectiv lui R4 (mesaj de tipul multicast pe IP-ul 224.0.0.6). In figura de mai jos, poti vedea cum transmite, R5, mesajul mai departe.

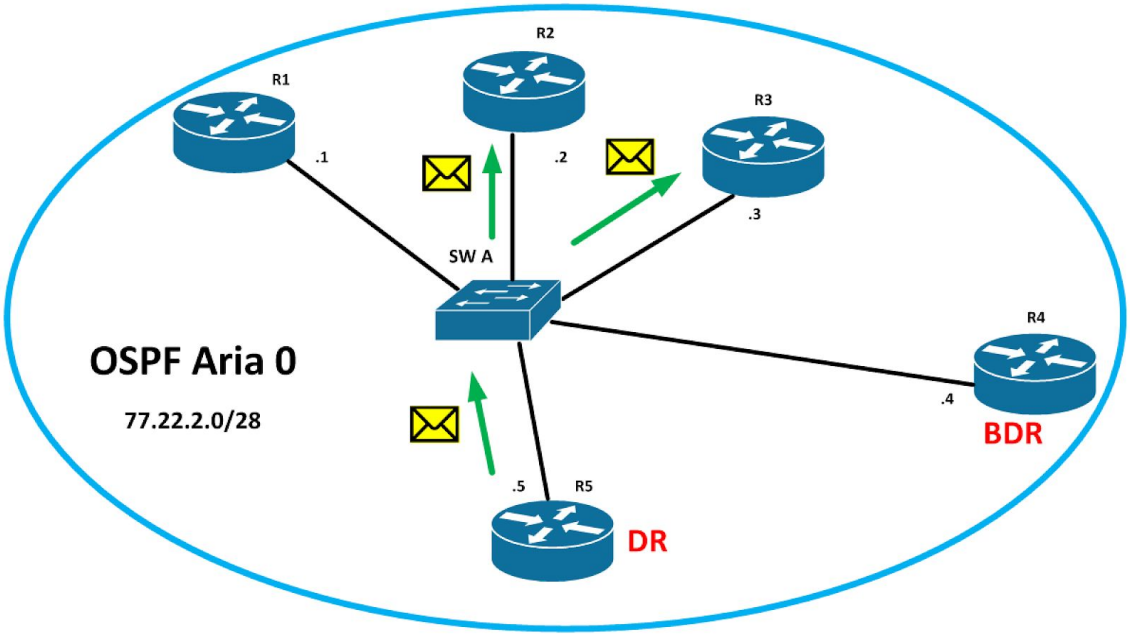


Figura 12.8

Pentru ca un Router sa ajunga DR, el trebuie mai intai sa fie ales. Procesul de selectie pentru DR/BDR este urmatorul:

1. ***Prioritate*** (by default are valoarea 1)
2. ***RID*** (Router ID)

Daca ***prioritatea*** *este la* ***egal*** pentru fiecare Router in parte, atunci **cel mai mare RID** va castiga si va fi ales DR, iar in urmatorul RID va fi ales BDR. **Router ID-ul (RID)** are rolul de a identifica in mod unic un Router in retea (mai exact in procesul OSPF). Daca a fost ales DR si BDR, restul Routerelor vor purta denumirea de **DROther**. *RID se stabileste astfel:*

- Setat **manual**, prin comanda R(router-config)#**router-id 1.1.1.1**
- Setat **automat**:
    - Cel mai mare IP al unei interfete Loopback
    - Cel mai mare IP al unei interfete active (daca nu exista Loopback)

Asdar, Routerul cu RID-ul mai mare va deveni DR, respectiv BDR. In scenariile de mai devreme, Routerele nu au avut configurate RID sau o adresa IP pe o interfata loopback, deci **RID-ul** a fost ales pe baza **celui mai mare IP** de pe o interfata activa:

R1 RID: 77.22.2.1
R2 RID: 77.22.2.2
R3 RID: 77.22.2.3
R4 RID: 77.22.2.4 (**BDR**)
R5 RID: 77.22.2.5 (**DR**)

Astfel, R5 a fost ales DR pentru ca are RID-ul cel mai mare, iar R4 a fost ales BDR pentru ca are al 2 lea cel mai mare RID.

Ce se intampla daca “DR-ul moare” ?

Acum sa presupunem ca s-ar intampla ce va cu **R5** si acesta **nu ar mai fi disponibil**, atunci **R4 ar deveni D**R, iar **R3** ar deveni **BDR**. Daca R5 ar reveni in retea, R4 si R3 ar continua sa fie DR, respectiv BDR, iar R5 ar deveni DROther pentru ca reteaua este stabila. Daca si R4 ar fi indisponibil, atunci R3 ar deveni DR, iar R5 ar deveni BDR.

Arii in OSPF

OSPF foloseste (in spate) un algoritm numit **SPF** (**S**hortest **P**ath **F**irst), care prin design-ul sau (modul de functionare) necesita destul de multe resurse (CPU si memorie RAM) - comparativ cu alti algoritmi. Datorita acestui fapt, daca numarul rutelor este mult prea mare, atunci procesarea si stocarea acestora va consuma foarte multe resurse. Astfel, cei care au conceput OSPF-ul s-au gandit sa foloseasca un design pe baza de arii.

*"Modul de functionare al OSPF-ului este **bazat pe arii**."*

Fiecare arie poate contine 1 sau mai multe retele (si Routere). Scopul acestor arii este de a reduce consumul de resurse - CPU si RAM - al fiecarui Router in parte si de a simplifica intreaga topologie de retea pentru acestea. "**Aria 0**" este o arie speciala si reprezinta "coloana vertebrala" (backbone-ul) retelei. Toate celelalte arii trebuie sa fie conectate (printr-un Router) la aria 0. Acest Router, este unul special in OSPF, pentru ca face parte din mai multe arii, el fiind numit **ABR** (**A**rea **B**order **R**outer).

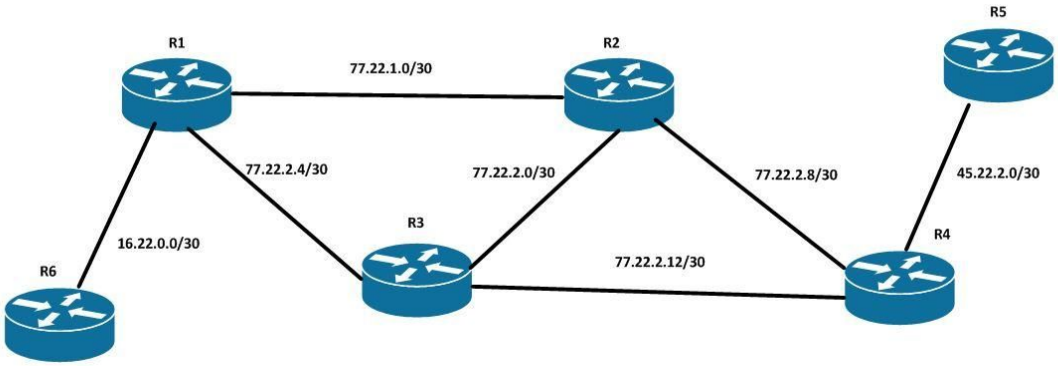


Figura 12.9

Topologia de mai sus cu design-ul in OSPF ar arata in felul urmator:

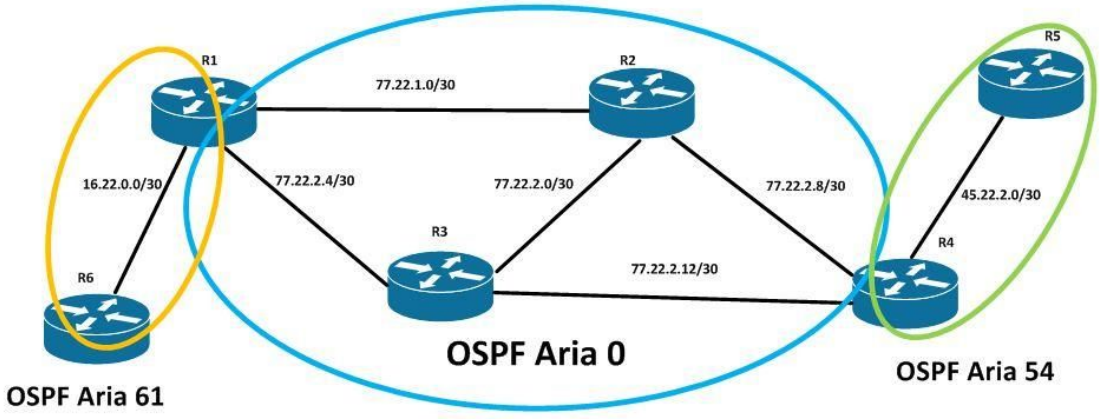


Figura 12.10

Fiecare Router va procesa si calcula doar informatiile de rutare (retele, starea / viteza link-urilor etc) din aria in care se afla. ABR-ul este Routerul special care face legatura intre aceste arii si in acelasi timp proceseaza si informatia de rutare din fiecare arie in parte.

Configurare OSPF cu o singura Arie (Single Area)

Sa presupunem ca avem reteaua din topologia de mai jos:

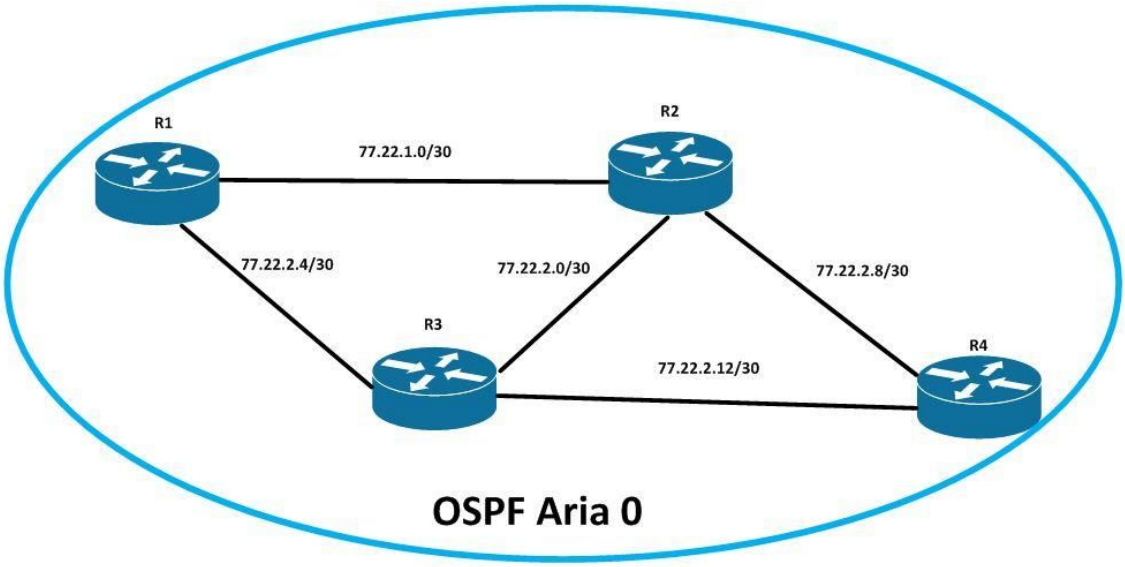


Figura 12.11

Scopul acestui exercitiu este a configura OSPF pe fiecare dintre Routerele din topologia de mai sus, astfel incat ele sa faca schimb de informatii de rutare (retele) pentru a ne permite conectivitate end-to-end. Toate aceste Routere se vor afla in Aria 0. Sa incepem config-ul cu Routerul R1:

```
R1(config)#router ospf 1
R1(router-config)#network 77.22.2.4 0.0.0.3 area 0
R1(router-config)#network 77.22.1.0 0.0.0.3 area 0
```

In cazul acesta, comanda "***router ospf 1***" *va porni procesul OSPF avand numarul 1* drept referinta. Dupa aceasta comanda, urmeaza sa includem retele in procesul de OSPF in aria specifica fiecarui retele in parte (conform design-ului).

Vom face asta folosind comanda "***network***" urmata de adresa de retea, wildcard mask (255.255.255.255 - masca) si numarul/id-ul ariei.

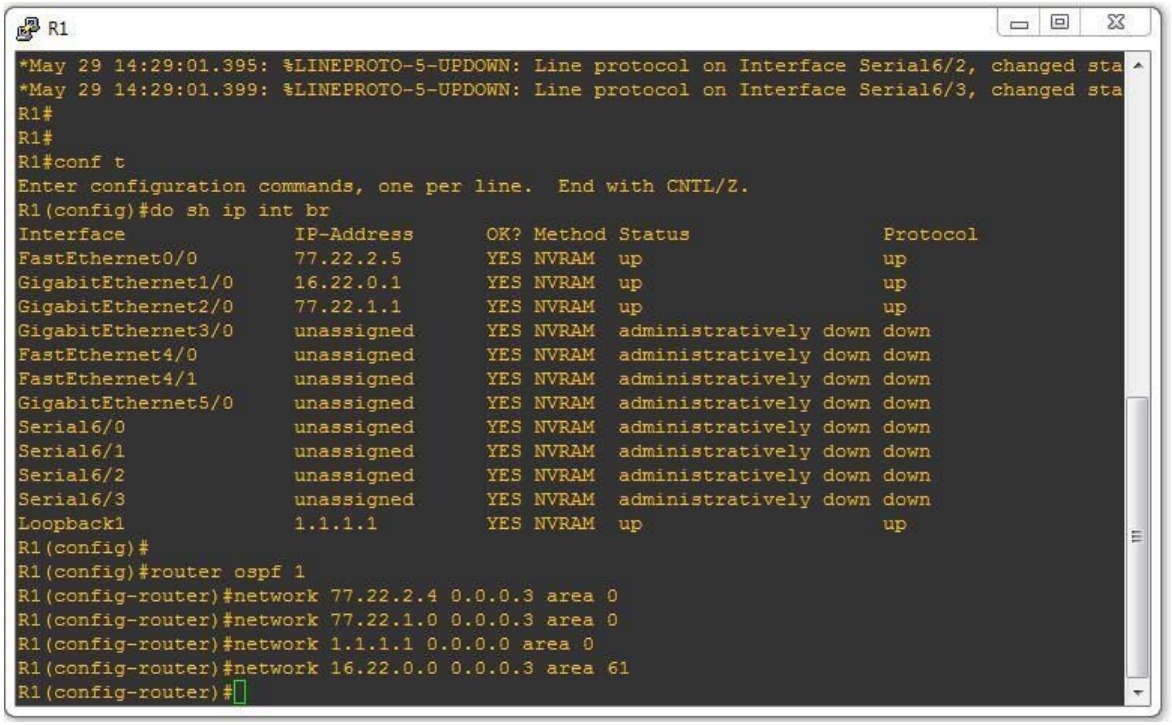


Figura 12.12

Practic, **noi pornim protocolul de rutare** pe interfetele specificate cu comanda "***network***".

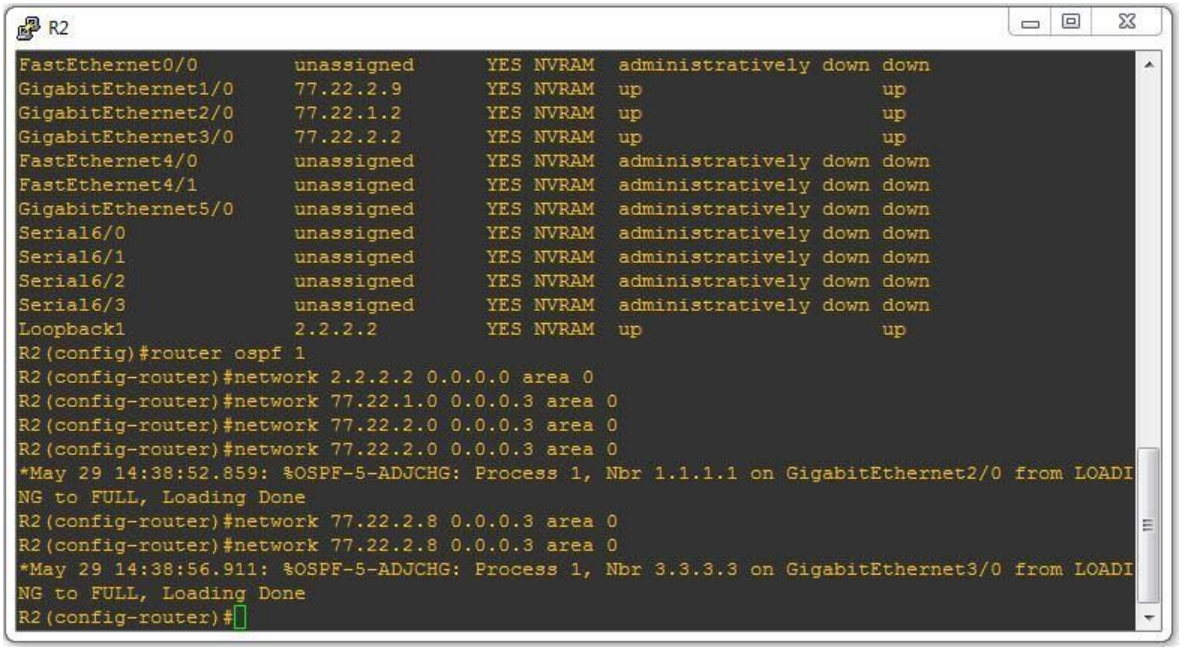


Figura 12.13

Pe R2:

```
R2(config)#router ospf 1
R2(router-config)#network 77.22.2.8 0.0.0.3 area 0
R2(router-config)#network 77.22.2.0 0.0.0.3 area 0
R2(router-config)#network 77.22.1.0 0.0.0.3 area 0
```

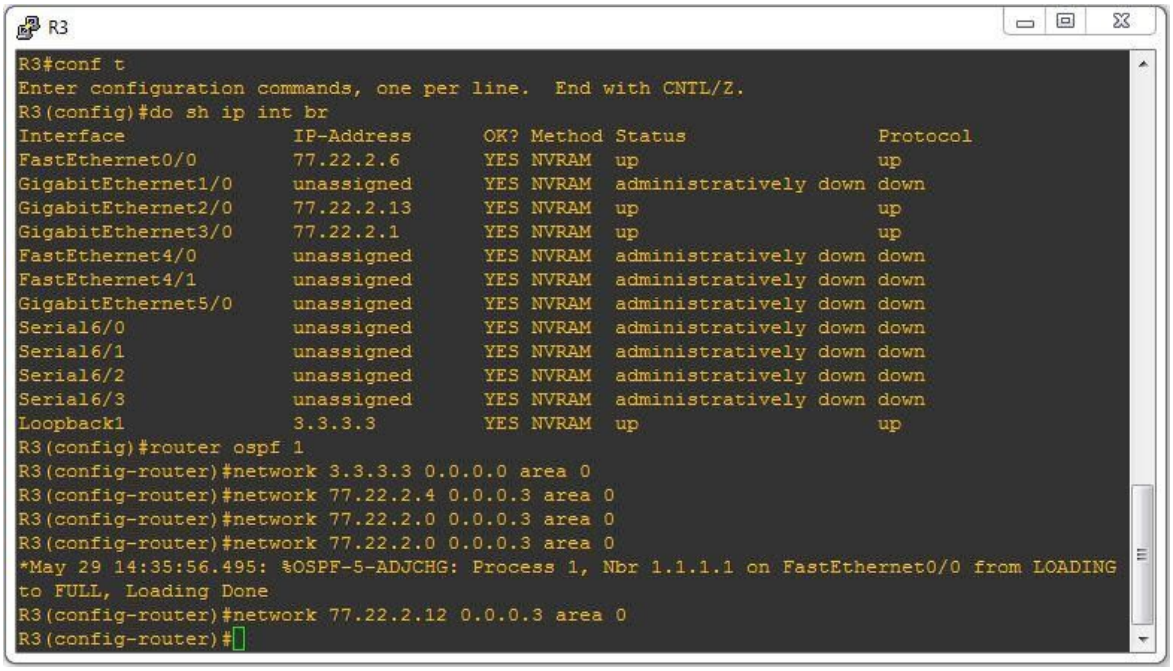


Figura 12.14

Pe R3:

```
R3(config)#router ospf 1
R3(router-config)#network 77.22.2.0 0.0.0.3 area 0
R3(router-config)#network 77.22.2.4 0.0.0.3 area 0
R3(router-config)#network 77.22.2.12 0.0.0.3 area 0
```

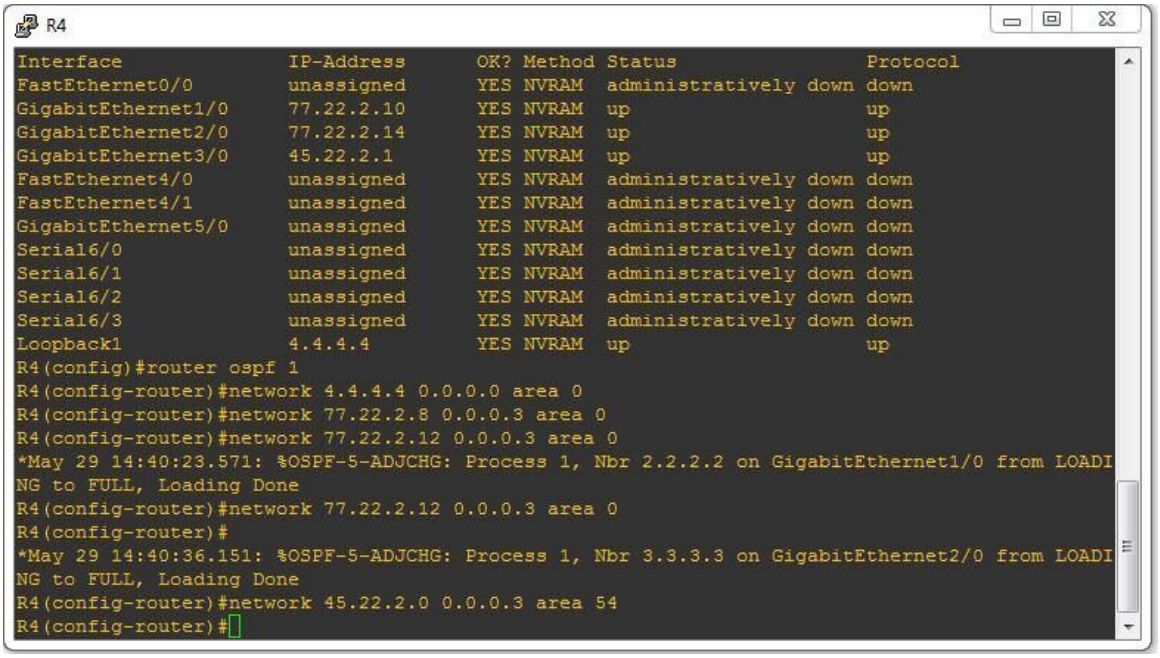


Figura 12.15

Pe R4:

```
R4(config)#router ospf 1
R4(router-config)#network 77.22.2.8 0.0.0.3 area 0
R4(router-config)#network 77.22.2.12 0.0.0.3 area 0
```

Astfel, am reusit sa configuram OSPF Area 0 pe toate cele 4 Routere. Dupa cum poti vedea in fiecare figura, dupa adaugarea comenzii network, au inceput sa apara anumite mesaje OSPF care ne anunta faptul ca o relatie de adiacenta a fost formata intre cele 2 Routere vecine. Iata cateva comenzi pe care le poti folosi pentru a verifica ca totul functioneaza cum ar trebui:

1. **#show ip route**
2. **#show ip ospf neighbors**
3. **#show ip protocols**

Configurare OSPF cu mai multe Arii (Multi-Area)

Acum sa presupunem ca reteaua a fost extinsa prin adaugarea a 2 Routere (R5 si R6):

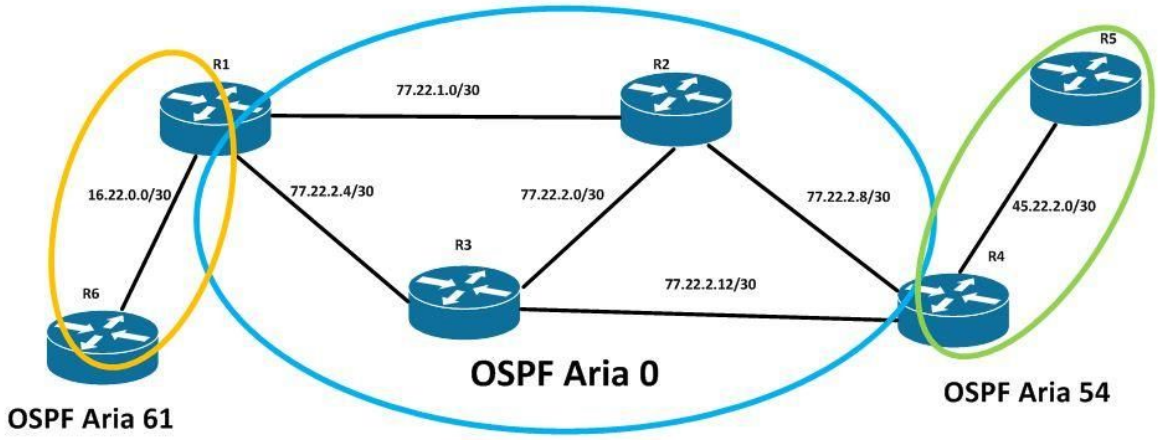


Figura 12.16

Acum, reteaua fiind mai mare, dorim sa schimbam design-ul (deoarece mai multe Routere inseamna mai multa informatie de procesat si mai multe resurse consumate - memorie RAM si CPU) si sa adaugam 2 arii: Aria 61 si Aria 54, ambele conectate printr-un Router special (numit **ABR**) la Aria 0. In aceasta topologie, **ABR va fi R1** (intre Aria 0 si 61), respectiv **R4** (intre Aria 0 si 54).

Setarile OSPF pentru Aria 61

Iata si configul pentru Routerul **R6** (care se afla impreuna cu R1 in aria 61):

```
R6(config)#router ospf 1
R6(router-config)#network 16.22.0.0 0.0.0.3 area 61
```

Iar pe **R1** vom adauga urmatoarea comanda:

```
R1(config)#router ospf 1
R1(router-config)#network 16.22.0.0 0.0.0.3 area 61
```

Aceste comenzi (pe R1 si R6) **leaga** cele 2 Routere in **aria 61** si conecteaza aceasta arie la Aria 0 (prin ABR-ul R1).

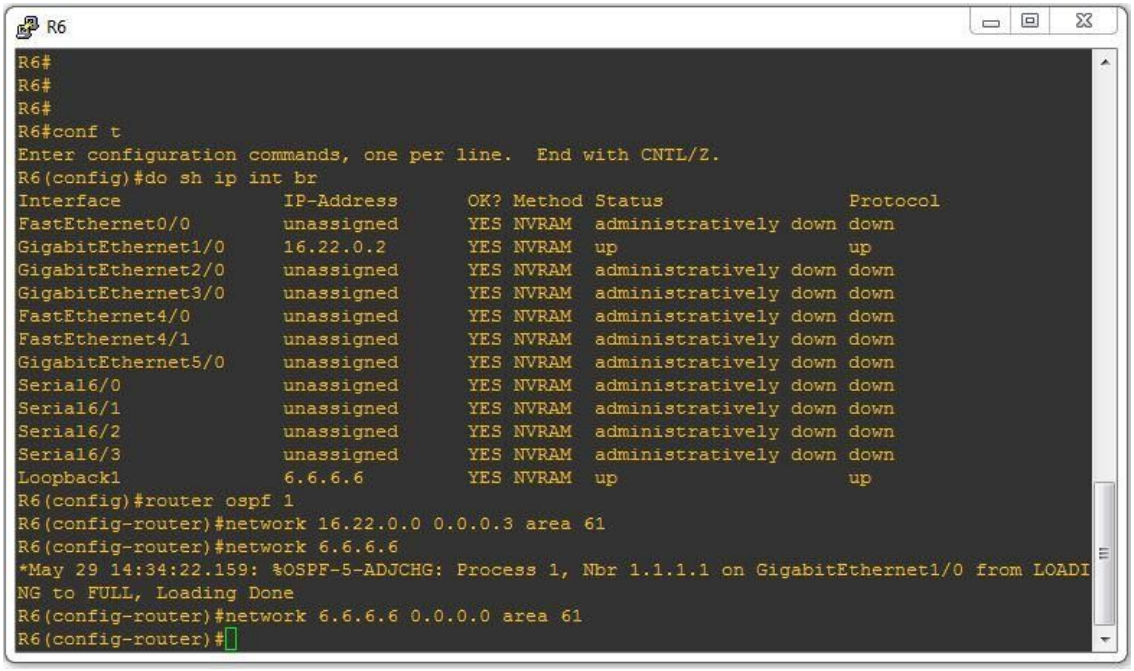


Figura 12.17

Setarile OSPF pentru Aria 54

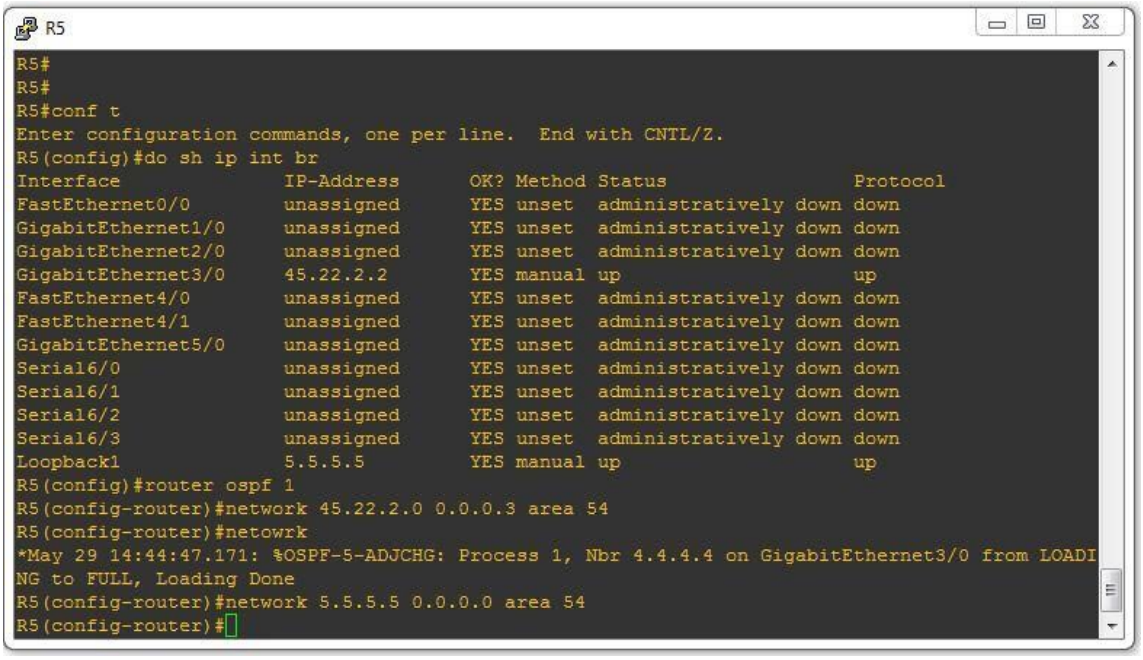


Figura 12.18

lata si configul pentru **R5**:

```
R5(config)#router ospf 1
R5(router-config)#network 45.22.2.0 0.0.0.3 area 54
```

Iar pe **R4** vom adauga urmatoarea comanda:

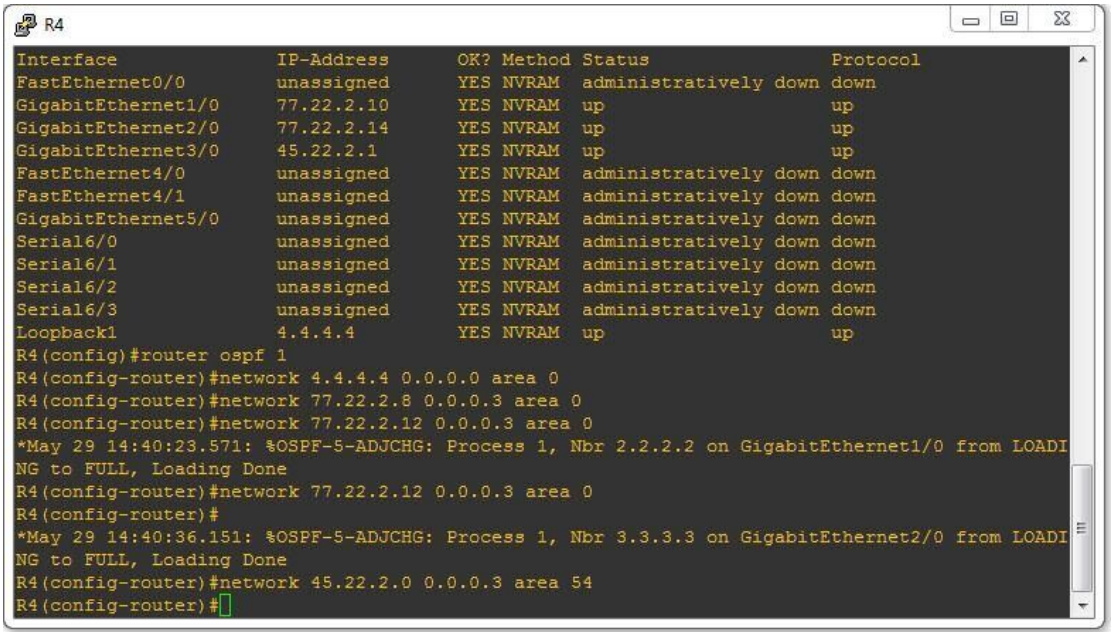


Figura 12.19

```
R4(config)#router ospf 1
R4(router-config)#network 45.22.2.0 0.0.0.3 area 54
```

Verificarea setarilor in OSPF

Iata cateva comenzi pe care le poti folosi pentru a verifica ca totul functioneaza cum ar trebui:

1. **#show ip route**
2. **#show ip ospf neighbors**
3. **#show ip protocols**
4. **#show ip ospf interface Gig1/1**
5. **#show ip ospf database**

Acum, in cele ce urmeaza, vom aplica aceste comenzi astfel incat sa vedem in mod clar ce am setat noi pe aceste Routere, sa vedem ce informatii de rutare au fost propagate si sa integelem mai bine modul de functionare al acestui protocol. Iata cum arata tabela de rutare de pe R3, respectiv R6:

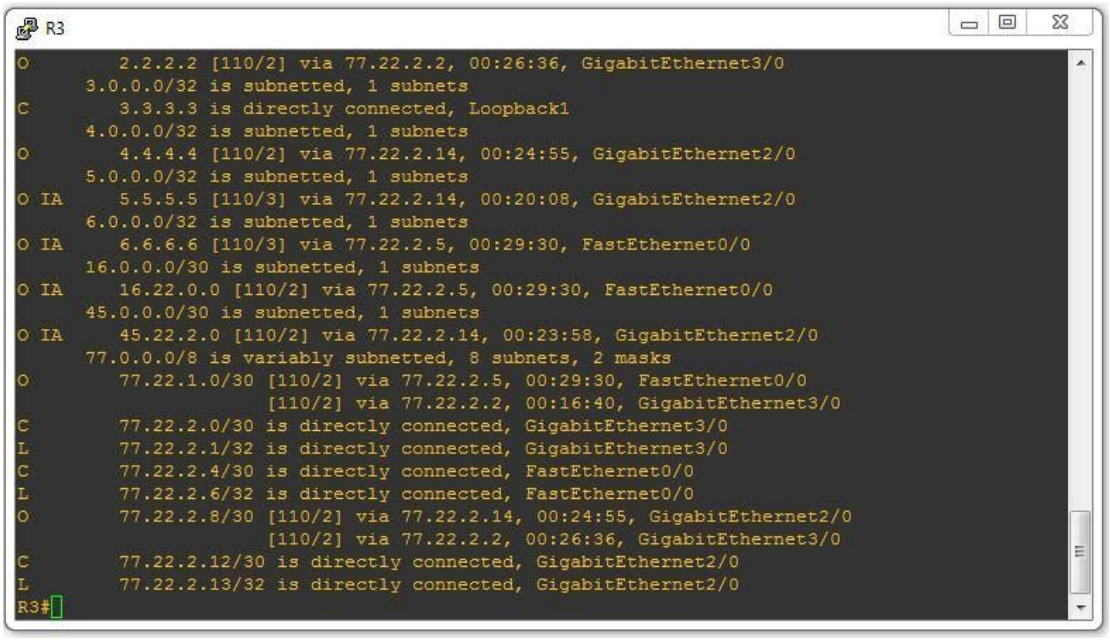


Figura 12.20

Dupa cum poti vedea, rutele care incep cu "**O**" reprezinta ca au fost primite prin OSPF **SI** *se afla in aceeasi arie* (in acest scenariu, aria 0). Celelalte rute care incep cu "**O IA**" au fost *originate din alte arii* (in acest scenariu fie din aria 61, fie din aria 54).

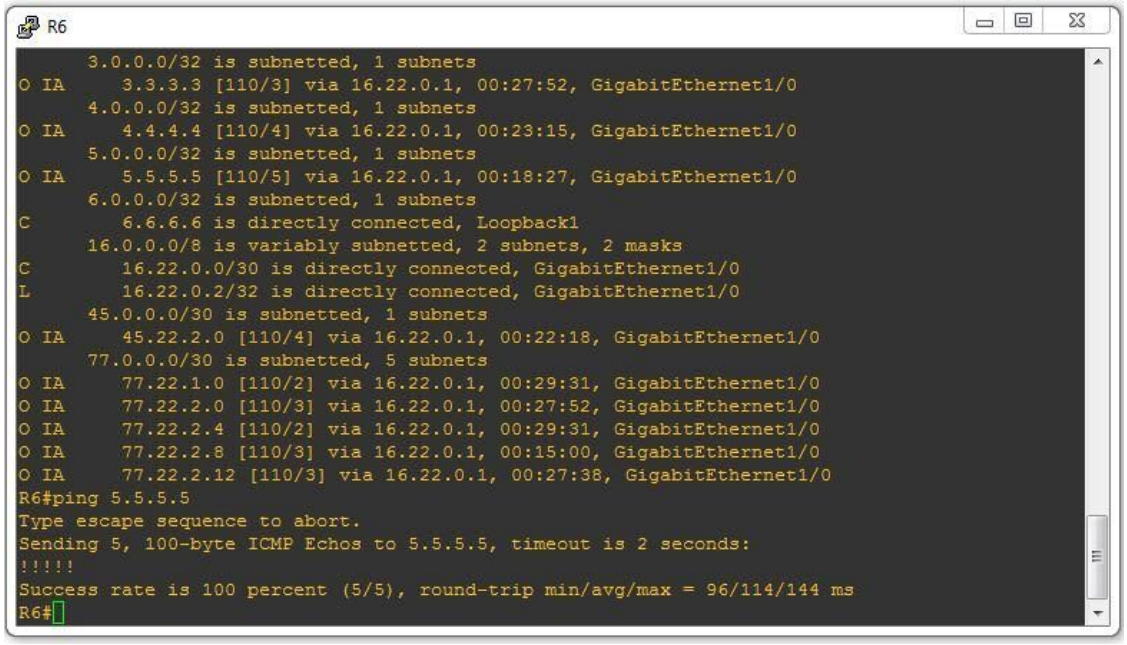


Figura 12.21

Tipuri de Arii

In OSPF exista mai multe tipuri de arii:

- *Backbone* - Aria 0
- *Standard*
- *Stub*
- *Totally Stubby*
- *NSSA (Not-So-Stubby Area)*
- *TS NSSA (Totally Stubby Not-So-Stubby Area)*

Scopul acestor *tipuri de arii* este **de a reduce informatia** care provine de la **Routerele** din **ariile vecine**. Spre exemplu, in figura de mai jos, din punctul de vedere al lui R5, **nu este** deloc **necesar** ca acesta *sa stie despre fiecare ruta* in parte (ex: Reteaua dintre R2-R1, reteaua dintre R3-R4, etc.) pentru ca acesta (indiferent de retea) are **o singura cale** (next-hop) spre acestea, prin R4.

In acest caz, **cea mai eficienta solutie** este de a seta o ruta statica default (0.0.0.0/0) pentru R5, respectiv R6. Aceste 2 routere poarta denumirea de Stub Routers, asadar aria poate fi considerata (setata): Stub.

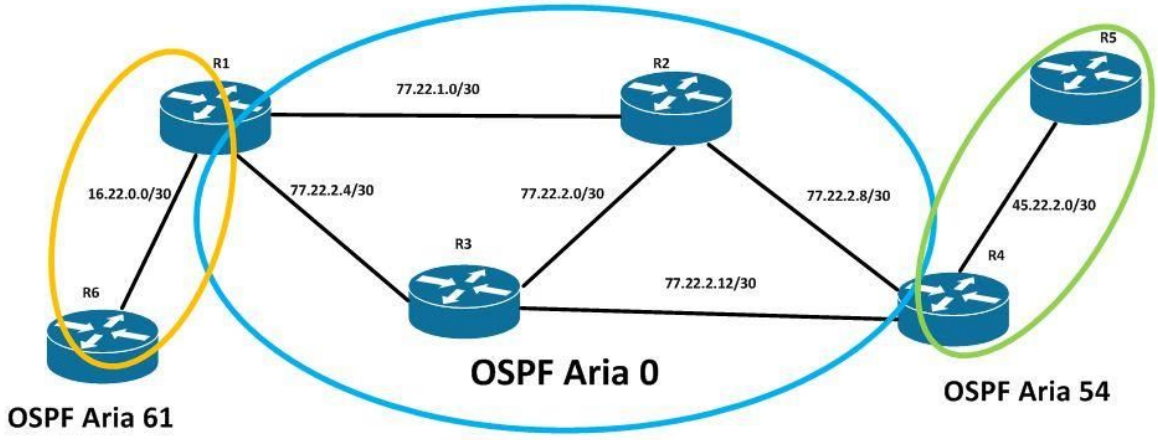


Figura 12.21

By default, Aria 0 este aria principala (***BACKBONE***) din design-ul OSPF. Ea este “core-ul” (miezul) retelei si face legatura cu celelalte arii. **Toate ariile trebuie sa fie conectate la Aria 0** (printr-un ABR). Orice arie diferita de 0, este by default de tipul ***STANDARD***. Tipul unei arii (Standard, Stub, NSSA etc.) indica informatia pe care o poate primi un Router (din acea arie).

1) O arie de tipul ***STUB*** *va bloca informatia care provine dintr-un protocol/*mediu **exteriorul** - diferit de OSPF (ex: EIGRP, RIP, rute statice etc.)

2) O arie de tipul **Totally Stubby** *va bloca orice informatie care provine dintr-o arie* exterioara (fata de cea din care face parte Routerul) sau din afara protocolului OSPF.

3) O arie de tipul **NSSA** (**N**ot **S**o **S**tubby **A**rea) este o arie speciala care permite ca, informatia (care provine din exterior-ul OSPF) sa ajunga la ABR cu scopul ca acesta sa o distribuie mai departe.

4) O arie de tipul **Totally NSSA** (**N**ot **S**o **S**tubby **A**rea) este la fel ca NSSA (permite informatia din exteriorul OSPF-ului sa tranziteze aria), doar ca informatia primita din alta arii este mult mai limitata (se foloseste o simpla ruta statica 0.0.0.0/0 catre ABR, dupa cum poti vedea si in exemplul de mai jos).

Configurare Tipurilor de arii in OSPF

Toate configuratiile pe care le-am facut pana acum au fost pentru arii de tipul Standard, dar acum vom schimba acest lucru mai ales pe cele 2 arii (54 si 61), diferite de aria 0.

NOTA: **Aria 0** nu-si poate modifica tipul. Ea este si poate fi doar aria principala (**“backbone”**)

In figurile de mai jos (pe topologia initiala), am configurat ***aria 54*** ca fiind **Stub** si ***aria 61*** ca fiind **Totally Stubby**.Sa incepem cu Aria 54 care este de tipul **Stub**. Iata si configul acesteia:

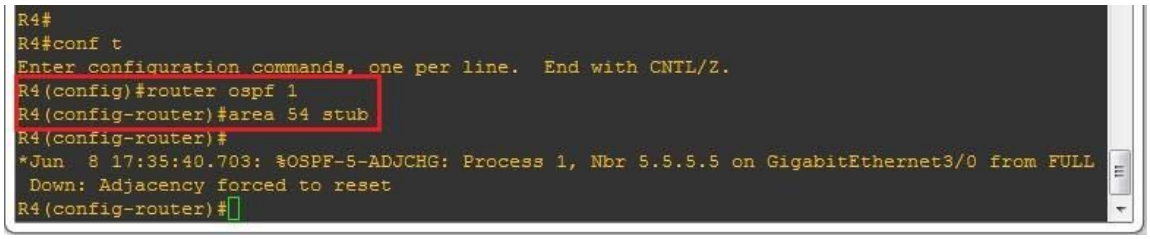


Figura 12.22

```
R4(config)#router ospf 1
R4(router-config)#area 54 stub
```

Dupa cum poti vedea, odata ce am configurat aria 54 ca fiind **Stub**, am pierdut adiacenta cu Routerul R5 pentru ca tipul ariilor (stub - pe R4 si standard - pe R5) nu corespund.

Dar odata ce am configurat si pe R5 ca fiind intr-o arie Stub, adiacenta s-a ridicat iar cele 2 Routere comunica (si schimba informatie de rutare) in mod dinamic. Acesta informatie de rutare este mult mai putina (in comparatie cu cea existenta initial). Exemplu: intr-o arie stub nu vor fi transmise informatiile de rutare provenite din exterior (retele diferite fata de cele ale OSPF-ului, ex: Rute Statice, EIGRP, RIP etc.). In tabela de rutare a lui R5 a aparut o ruta default (0.0.0.0/0) trimisa automat de catre ABR (R4).

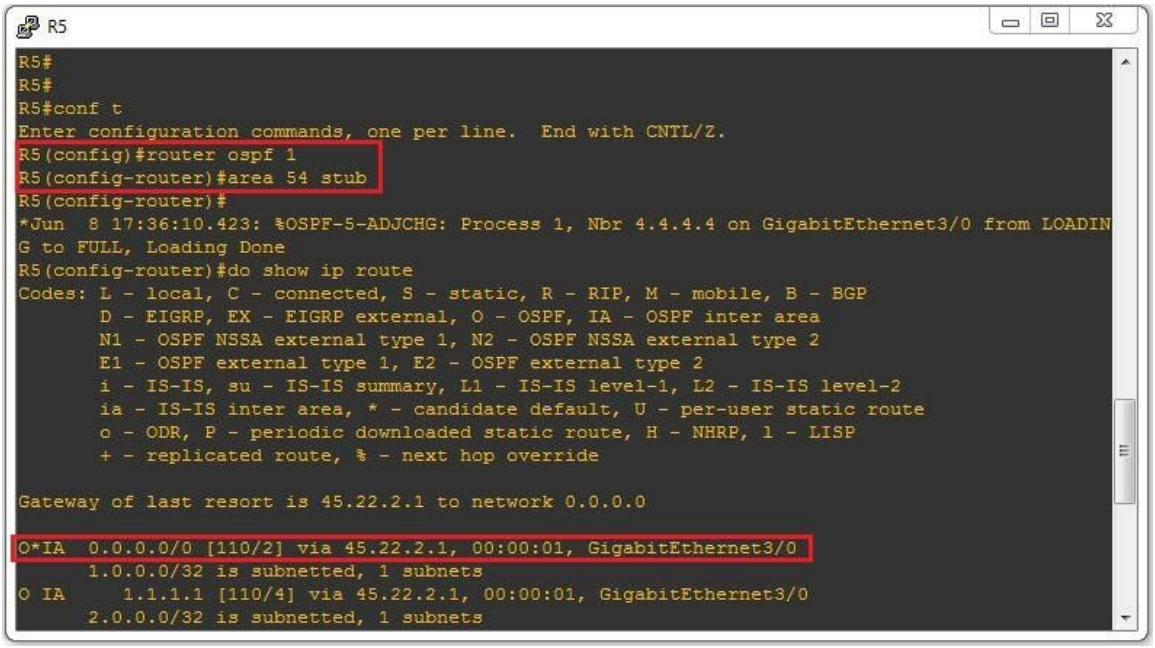


Figura 12.23

```
R5(config)#router ospf 1
R5(router-config)#area 54 stub
```

Acum sa trecem la urmatorul de tip de arie, **Totally Stubby**, si vom demonstra functionalitatea ei prin exemplul de mai jos.

```
R1(config)#router ospf 1
R1(router-config)#area 61 stub no-summary
```

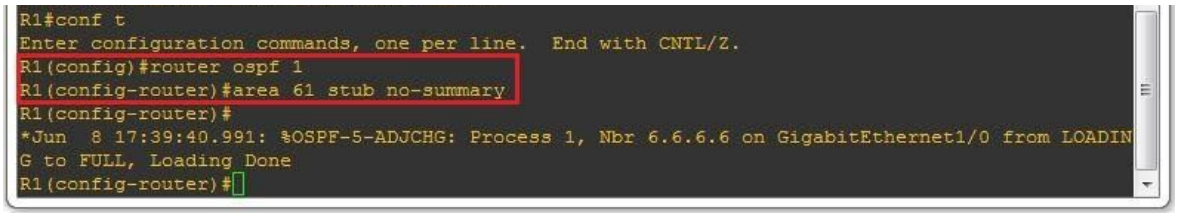


Figura 12.24

Aria 61 a fost configurata ca fiind **Totally Stubby**, pe Routerele R1 si R6 iar rezultatul il putem vedea in figura de mai jos (prin output-ul comenzii ***#show ip route***, de pe R6). Configuram si pe R6 Aria ca fiind T.S.:

```
R6(config)#router ospf 1
R6(router-config)#area 61 stub no-summary
```

Dupa cum poti vedea (in figura de mai jos), toate rutele din OSPF au fost inlocuite cu o simpla ruta default 0.0.0.0/0 catre R1 (ABR-ul ariei 61). Practic, pentru R6 nu este necesar ca acesta sa stie fiecare ruta din intreaga topologie pentru ca acesta are un singur punct de iesire prin care poate trimite traficul (acesta fiind R1).

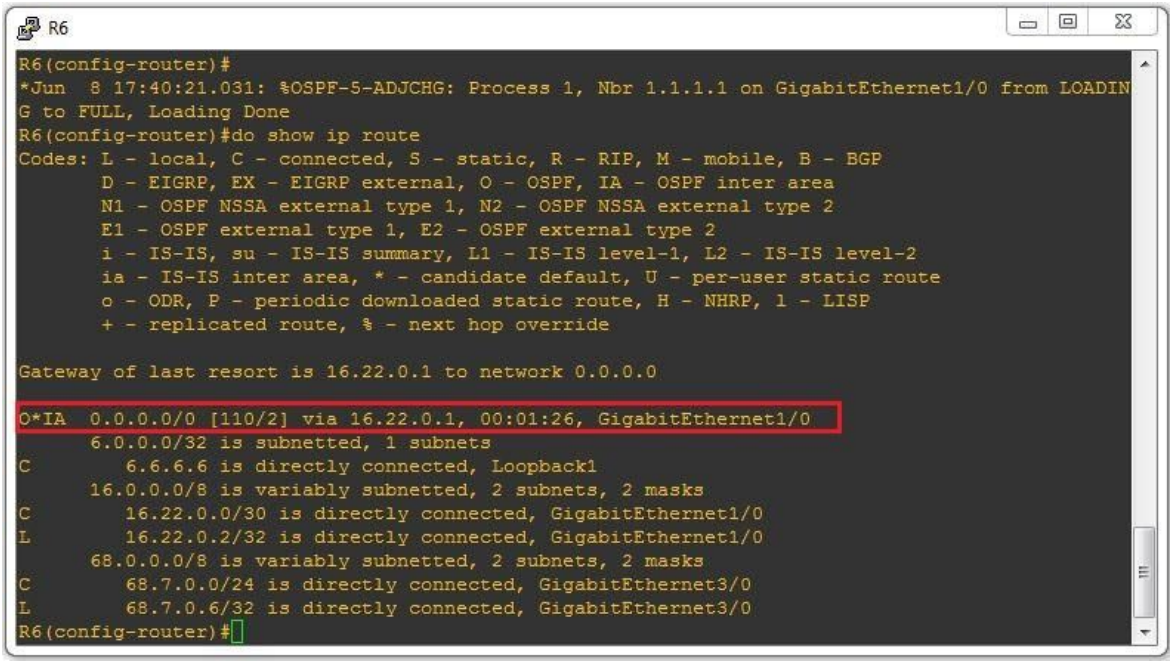


Figura 12.25

Verificarea Configuratiei

Dupa cum poti vedea in fiecare figura de mai sus, odata cu adaugarea retelelor (prin comanda *#network*) **incep sa se formeze adiacentele** dintre Routere. Acum a venit momentul sa verificam si sa vedem cum arata acestea.
Iata cateva comenzi utile:

- **#show ip ospf neighbors**
- **#show ip route**
- **#show ip protocols**
- **#show ip ospf interface Gi0/1**
- **#show run | section ospf**

`R1#show ip ospf neighbors`

```
R1#
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
3.3.3.3           1   FULL/DR         00:00:38    77.22.2.6       FastEthernet0/0
2.2.2.2           1   FULL/DR         00:00:22    77.22.1.2       GigabitEthernet2/0
6.6.6.6           1   FULL/DR         00:00:30    16.22.0.2       GigabitEthernet1/0
R1#
```

Figura 12.26

In outputul acestei comenzi putem vedea toti vecinii din OSPF (cei direct conectati cu R1). Putem vedea RID-ul lor (Router ID-ul), starea in ca se afla DR, BDR, DROTHER, sincronizarea bazelor de date (in cazul acesta este FULL), adresele IP ale Routerelor vecine etc.

`R1#show ip route`

```
R1
R1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback1
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 77.22.1.2, 00:03:49, GigabitEthernet2/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/2] via 77.22.2.6, 00:03:49, FastEthernet0/0
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/3] via 77.22.2.6, 00:03:39, FastEthernet0/0
                 [110/3] via 77.22.1.2, 00:03:39, GigabitEthernet2/0
      6.0.0.0/32 is subnetted, 1 subnets
O        6.6.6.6 [110/2] via 16.22.0.2, 00:03:59, GigabitEthernet1/0
 --More--
```

Figura 12.27

Aici putem vedea tabela de rutare si poti observa rutele cu O (indica faptul ca rutele au fost invatate prin OSPF). Pe langa asta poti observa in [110 (AD) / 2 (Metrica - aka **Cost**).

`R1#show ip protocols`

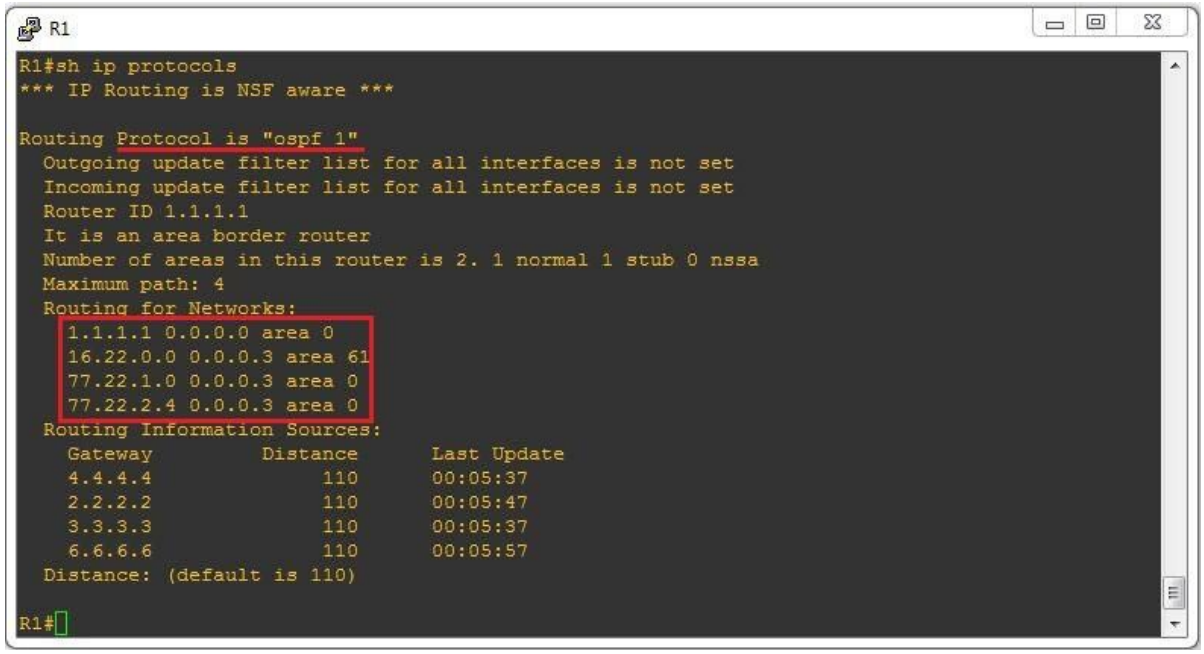


Figura 12.28

Cu acesta comanda putem vedea exact retelele in ce arii se afla, routerele vecine, RID-ul precum si alte informatii care tin de protocolul OSPF.

`R1#show ip ospf  interface Gi2/0`

Mai jos, poti vedea mai multe detalii care tin de configul OSPF-ului pe o anumita interfata.

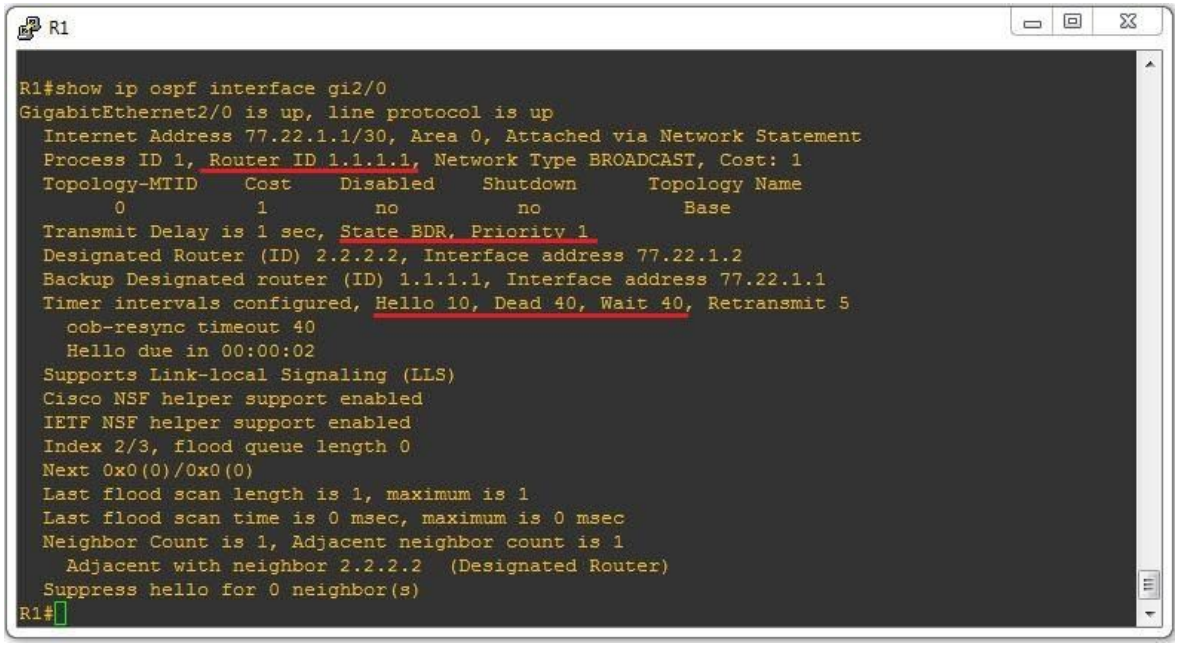


Figura 12.29

```
R1#show run | section ospf
```

Figura 12.30

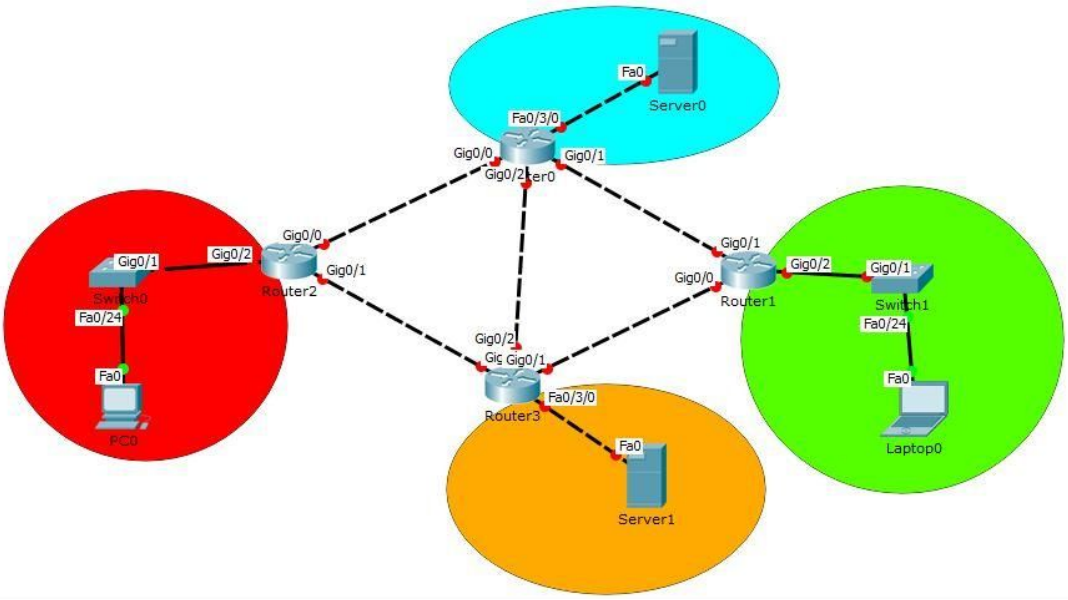
In figura de mai sus putem vedea exact comenzile pe care le-am dat pe R1. In cazul in care a intervenit o greseala in configurare (un misconfiguration), putem merge in #**router ospf 1** si aplica aceeasi comanda, dar cu #**no** in fata pentru **stergerea** ei.

Laboratorul #6

Acum am ajuns la partea de laborator (partea practica), pe care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurari de baza cu OSPF in Area 0

Aplicam protocolul OSPF pe aceeasi topologie ca cea din laboratorul #5, dar focusul va fi pe **configurarea** si **verificarea** protocolului **OSPF**.

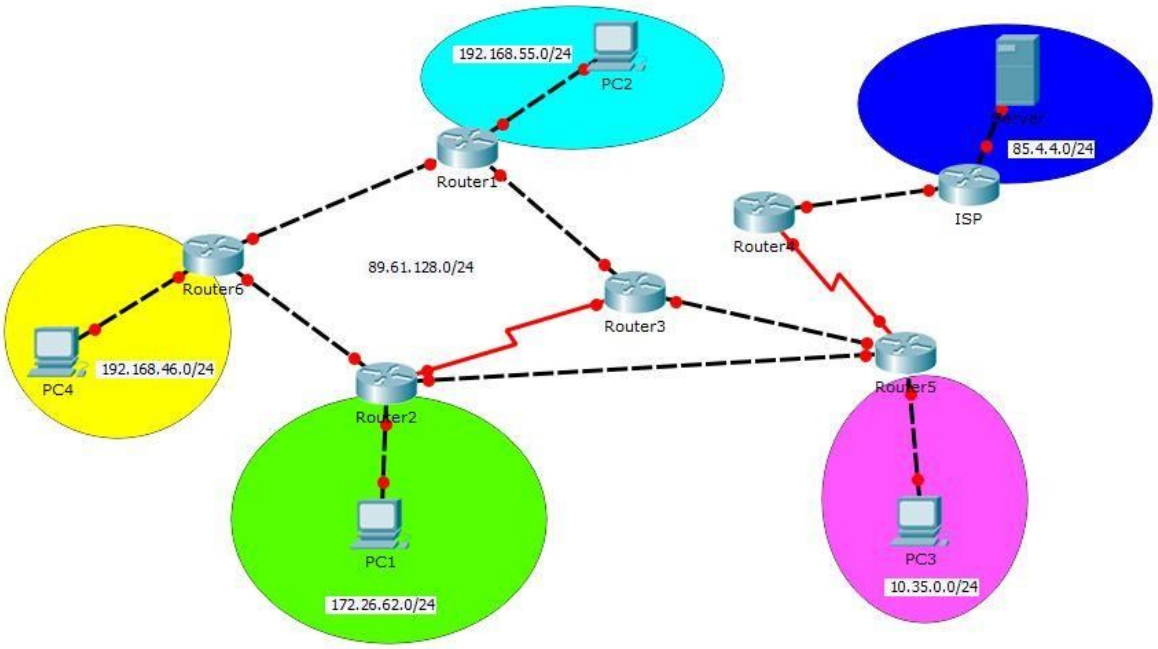


SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Laboratorul #7

Am ajuns iar la partea de laborator (partea practica), pe care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurari avansate cu OSPF Multi-Area



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Capitolul 13 - Advanced Distance Vector - EIGRP (Enhanced Interior Gateway Routing Protocol)

Acum sa facem trecerea de la protocoalele de rutare de tip Link-State (OSPF) si sa vorbim despre un protocol care este si Link-State, si Distance Vector. EIGRP este un protocol de rutare dezvoltat de Cisco (este **proprietary** si functioneaza **DOAR** pe echipamente **Cisco**). Dupa cum spuneam si mai devreme, el este de tipul Advanced (Hybrid) Distance Vector, adica imprumuta caracteristici atat de la Distance Vector cat si de la Links-State.

De ce EIGRP este Hybrid Distance Vector ?

EIGRP imprumuta urmatoarele *caracteristici* de la un protocol **Link-State**:

- Tabela de adiacenta cu vecinii
- O Tabela separata (de cea de rutare) pentru topologie
- Schimbul de mesaje de tip **Hello**

Iar de la **Distance Vector** aceste *caracteristici*:

- Se bazeaza pe informatia primita de la vecini (pentru a calcula cea mai buna cale)
- Nu mentine o harta activa cu fiecare cale existenta din retea

Cum Functioneaza EIGRP ?

Vom incepe sa vedem cum functioneaza acest protocol prin descrierea, pe scurt, a catorva elemente cheie care tin de el. Vom vorbi despre Metrica, AD (distanta administrativa), tipurile de mesaje, algoritm (sau modul de functionare) iar la sfarsit vom vedea cum il putem configura.

Metrica in EIGRP

Spre deosebire de celelalte protocoale, pentru EIGRP **calculul metricii este mai complex**. Daca la RIP aveam *numarul de hop-uri*, la EIGRP avem urmatoarele:

- *BW* - latimea de banda a interfetei - K1
- *Delay* - latenta interfetei - K2
- *Reliability* - increderea pe care o avem in interfata - K3
- *Load* - incarcarea interfetei - K4
- *MTU* - K5

Iata formula de calcul a metricii asa cum apare pe site-ul Cisco (https://goo.gl/2b6Yeb).

Metric = ([K1 * BW + (K2 * BW) / (256 - load) + K3 * delay] * [K5 / (reliability + K4)]) * 256

Fiecare dintre aceste K-uri se numesc **K-values** si pot avea valoarea 1 sau 0. **By default,** EIGRP foloseste doar **K1 si K3** (adica acestea **au valoarea 1**, iar **restul au valoarea 0**).

K1 = 1, K2 = 0, K3 = 1, K4 = 0, K5 = 0

Astfel formula pentru metrica va arata in felul urmator:

Metric = ([K1 * BW + K3 * delay] * 256

Sau poate fi simplificata si mai mult:

Metric = (BW + delay) * 256

Mult mai simplu, nu ? :)

Distanta Administrativa

Dupa cum am discutat si la RIP si OSPF, distanta administrativa reprezinta nivelul de incredere pe care un Router il are in sursa (protocolul) din care provine informatia despre o anumita retea.
Aceasta avea valoarea:

- 120 - RIP
- 110 - OSPF
- **90 - EIGRP**

Asadar, daca informatia despre o retea (ex: 10.0.0.0/8) provine din 2 surse, una fiind EIGRP iar cealalta fiind OSPF, Routerul o va alege pe cea din EIGRP deoarece **Distanta Administrativa este mai mica** (90 vs 110).

Algoritmul EIGRP - DUAL

Algoritmul folosit este DUAL (**D**iffusing **U**pdate **AL**gorithm) si este unul special dezvoltat de Cisco pentru a-l face pe EIGRP extrem de rapid si de eficient (din punct de vedere al resurselor necesare). Iata cativa dintre termenii cu care ne vom intalnii destul de des de acum in colo:

- *FD* - **Feasible Distance**
    - metrica catre o retea destinatie
- *S* - **Succesor**

- next-hop-ul catre o anumita retea destinatie
- *FS* - **Feasible Succesor**
  - un potential next-hop (printr-o cale de backup)
- *FC* - **Feasibility Condition**
  - conditia pe care trebuie sa o indeplineasca un Router pentru a deveni FS
- *RD/AD* - **Reported/Advertised Distance**
  - metrica pe care un Router (vecin) o are pana la destinatie

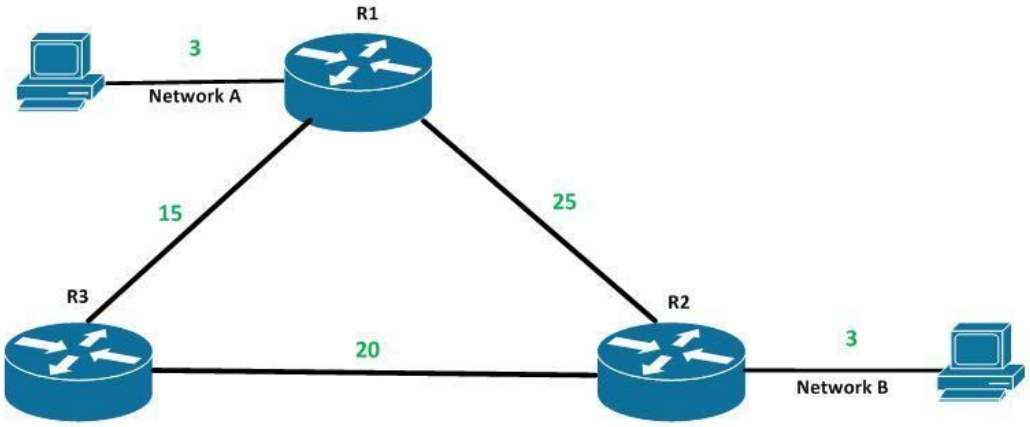


Figura 13.1

Sa luam ca exemplu topologia din figura de mai sus si sa presupunem ca R1 trebuie sa determine cea mai buna cale pana la reteaua B. Acesta are 2 posibilitati: una prin R2, cealalta prin R3.

Prin **R2** metrica va fi: 25 + 3 = **28**

Prin **R3** metrica va fi: 15 + 20 + 3 = **38**

La fel ca si la OSPF si RIP, ruta cu **cea mai mica metrica** este aleasa, in acest scenariu, ca fiind cea prin R2 (cu metrica 28).

Asadar, **R2** va purta denumirea de **Succesor (S)**, **Metrica** (28) se va numi **Feasible Distance (FD)**, iar metrica (3) de la R2 la reteaua B va fi **Reported Distance (RD)** pentru R1. Acum, R1 a stabilit cea mai buna cale pana la destinatia B, acesta fiind prin R2. Avantajul pe care il ofera algoritmul **DUAL** (fata de orice alt protocol de rutare) este faptul ca R1 isi poate “pre-calcula” o ruta de backup (folosind FC), in cazul in care prima ruta nu mai este valabila. In acest caz **Feasibility Condition** (**FC**) spune asa:

*Daca exista o cale secundara (fata de cea principala: R1 -> R2 -> B) catre o retea destinatie, EIGRP o va lua in considerare **doar daca** metrica (23) de la R3 pana la destinatie (B) este mai mica decat metrica (28) rutei principale (R1 -> R2 -> B).*

Cu alte cuvinte, metrica (rutei secundare) de la R3 -> R2 -> B (care este **23**) trebuie sa fie mai mica decat metrica rutei principale (R1 -> R2 -> B), care este **28**.Astfel, in aceasta situatie 23 < 28, deci (pentru R1) ruta prin R3 va fi considerata "**Backup Route**", iar R3 va fi considerat **Feasible Succesor (FS)**, cu metrica totala de 38 (15 + 23).

De ce sunt toate acestea importante ? Pentru ca, EIGRP, odata ce este stabilita o **ruta de backup**, in cazul unei defectiuni in retea (si pierderea rutei principale), **NU va trebui sa recalculeze o alta ruta pana la destinatie**. Astfel **down time-ul** din retea va fi aproape inexistent (versus cealalta alternativa in care ar fi trebuit sa trimita mesaje speciale de Query catre Routerele vecine pentru a afla daca mai exista o alta cale spre destinatie, astfel *pierzandu-se multe secunde care pot impacta grav reteaua*). EIGRP stocheaza informatia (atat cea primita de la vecini cat si cea generata local) in 3 tabele diferite:

1) **Tabela de adiacenta**
    a) Contine informatii despre Routerele vecine direct conectate
2) **Tabela de topologie**
    a) Stocheaza **informatia primita de la vecini** si este folosita (de catre algoritmul DUAL) pentru a calcula cea mai buna cale pana la destinatie (si inclusiv rutele de backup)
3) **Tabela de rutare**
    a) Stocheaza **ruta** cu **cea mai buna cale** pana la destinatie (**filtrata** de catre DUAL din tabela de topologie)

Hai sa rezumam ce am discutat mai sus:

- Ruta **R1 -> R2 -> B** este ruta principala (stocata in tabela de rutare), unde R2 este **S** (vezi notatia de mai sus) pentru R1, iar **FD-ul** va fi 28.
- Ruta **R1 -> R3 -> R2 -> B** este ruta secundara (de backup) pentru ca **FD-ul** de la R3 -> R2 -> B **este mai mic** (**23**) fata de **FD-ul** de la R1 -> R2 -> B (**28**). Astfel FC-ul spune ca ruta prin R3 poate fi considerata o "ruta de backup" si va fi stocata in tabela de topologie

Toate aceste informatii le putem gasi in **tabela de topologie** a EIGRP-ului:

De exemplu, ruta 10.0.0.0/24 are un *succesor* (12.92.2.2) cu metrica (FD) 130816.

P-ul din fata vine de la Passive si se refera la faptul ca aceasta ruta este pasiva (adica algoritmul EIGRP-ului, DUAL, nu face o cautare activa a acestei rute). **Passive** is good (ruta este stabila). **Active** is bad (Routerul cauta activ o alta cale catre aceasta ruta pentru ca cea initiala a fost pierduta)

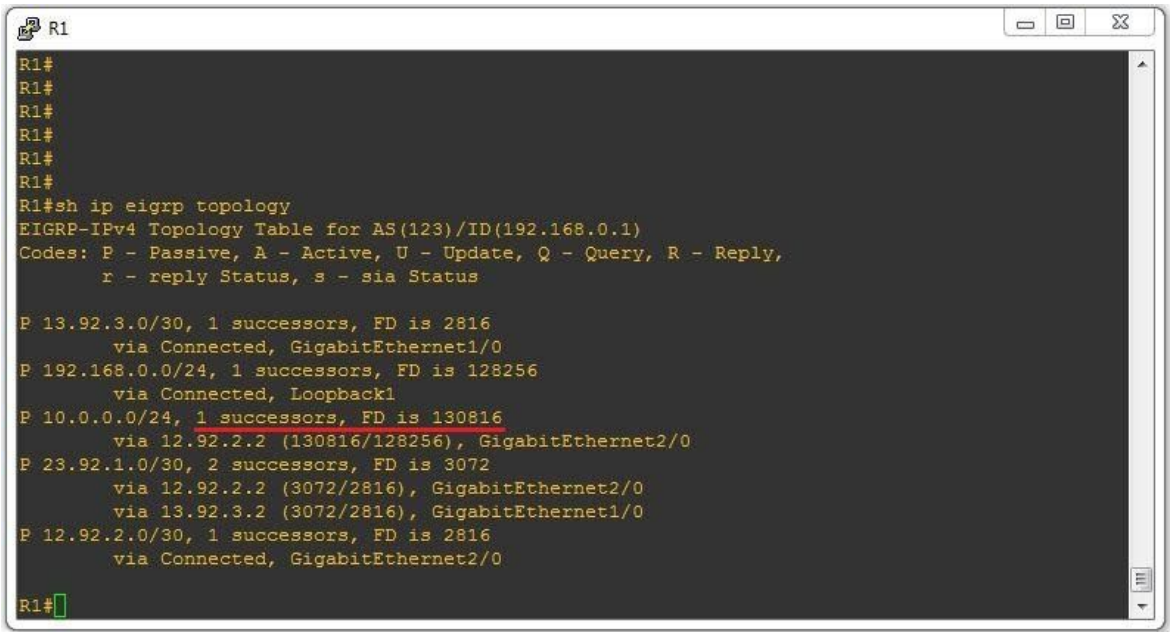


Figura 13.2

Tipuri de Mesaje in EIGRP

Similar cu OSPF, si EIGRP contine mesaje specifice pe care le foloseste pentru crearea adiacentelor si schimbul de informatii intre Routere. Acestea sunt:

- *Hello*
- *Update*
- *Acknowledge*
- *Reply*
- *Query*

1) Mesajele de Tip Hello

In EIGRP, mesajele *Hello* sunt **folosite** pentru **a forma adiacenta** intre Routere si sunt trimise la fiecare **5 secunde**. Daca nu sunt primite 3 Hello-uri consecutive, EIGRP va rupe adiacenta si va scoate din tabela de rutare retelele invatate.

- *Hello Interval = **5** secunde*
- *Hold Interval = **15** secunde*

Aceste Hello-uri sunt trimise pe adresa destinatie **224.0.0.10** (asa cum poti vedea si in figura de mai jos):

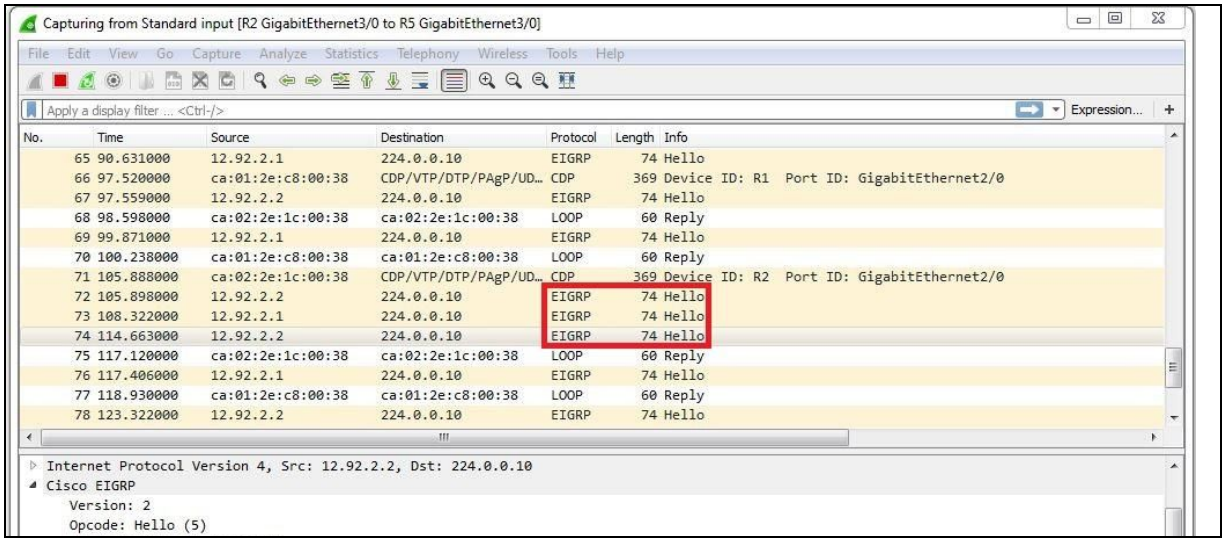


Figura 13.3

2) Mesajele de Tip Update & ACK

Spre deosebire de RIP (si similar cu OSPF), EIGRP trimite **doar update-uri necesare** (ala update-uri partiale) Routerelor vecine. Dupa cum poti vedea si in captura Wireshark de mai jos, sunt trimise mesajele de tip Update dupa care se va trimite un ACK de la vecin.

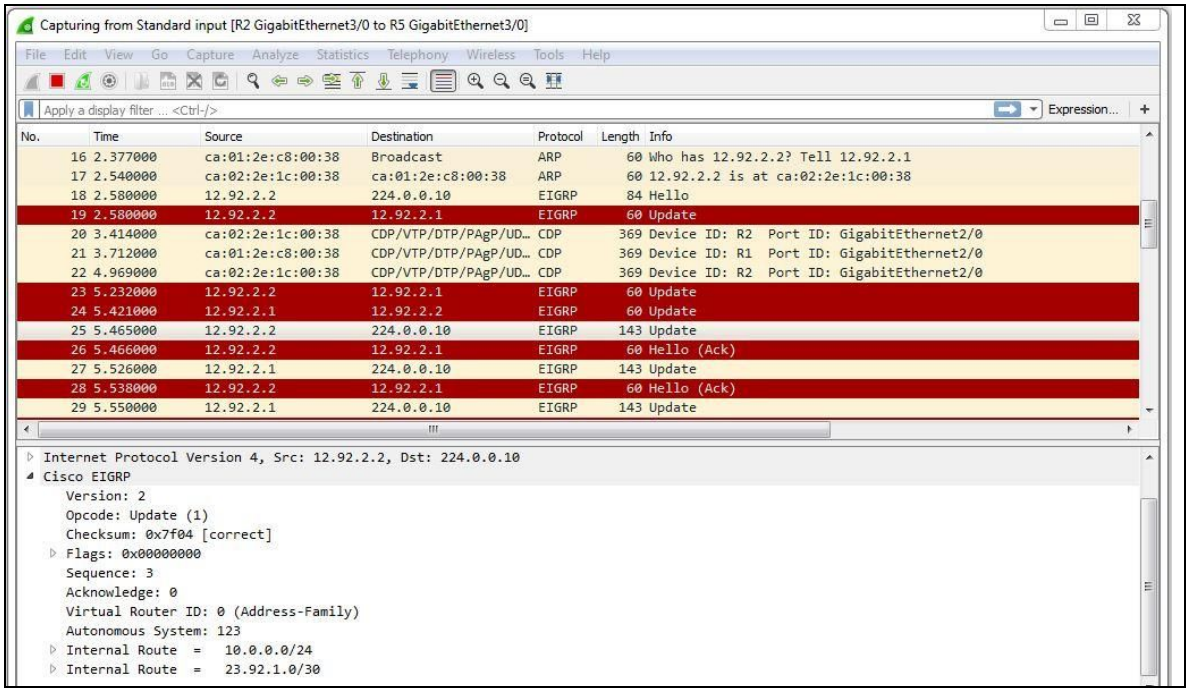


Figura 13.3

3) Mesajele de Tip Query & Reply

In momentul in care se **intampla ceva cu o ruta** din retea (acesta “cade”), Routerele “constiente” de acest eveniment incep **sa-si intrebe vecinii** (prin mesaje de tip **Query**) daca stiu ceva legat de acele mesaje. Iar in acest scenariu avem 2 posibilitati:

1. In cazul in care vreun Router **stie de acea retea**, va trimite un mesaj de tipul **Reply**, care se va propaga in intreaga retea. Fiecare Router in partea **va adauga noua** cale catre acea retea in tabela de rutare.
2. In cazul in care nici un Router **nu stie de acea retea**, Routerele vor genera *mesaje de tipul* **Reply** cu acea retea, dar cu o metrica infinita (insemnand ca nu stiu de acea retea si nu au idee cum sa ajunga la ea). Astfel reteaua va fi scoasa din tabela de rutare.

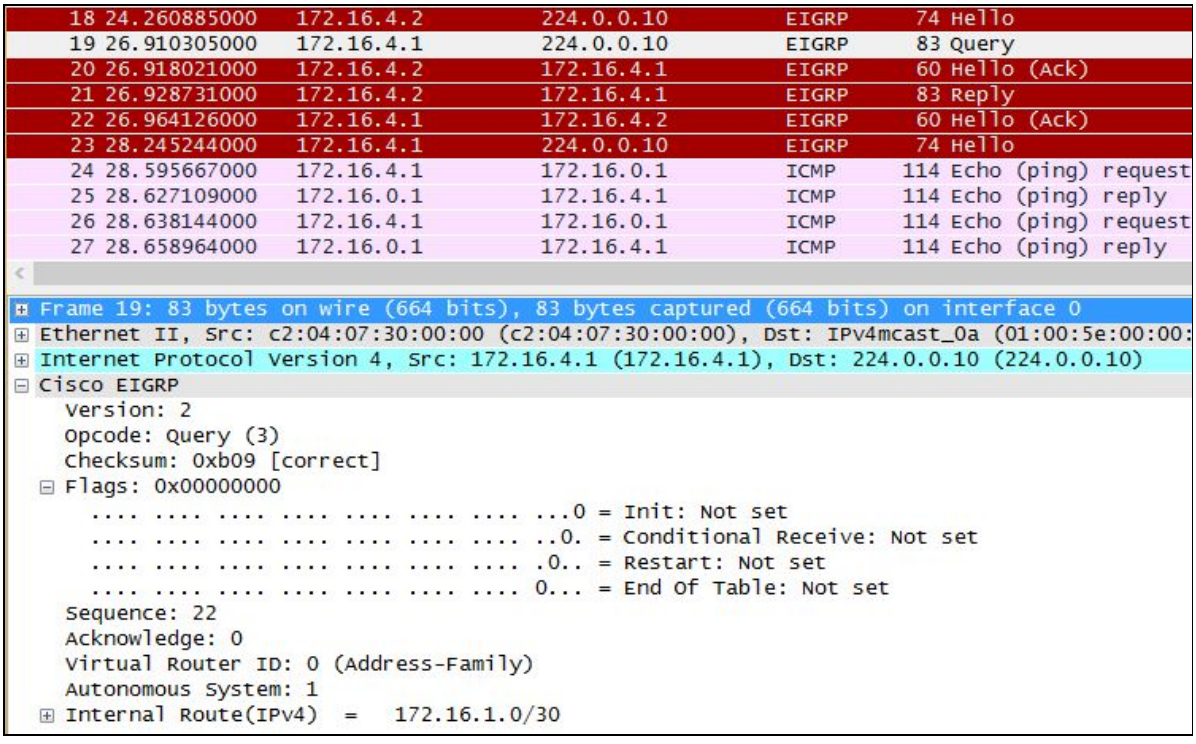


Figura 13.4

Cum Configuram EIGRP pe Routere ?

Acum sa trecem la partea de configurare a EIGRP-ului si sa luam ca exemplu topologia de mai jos:

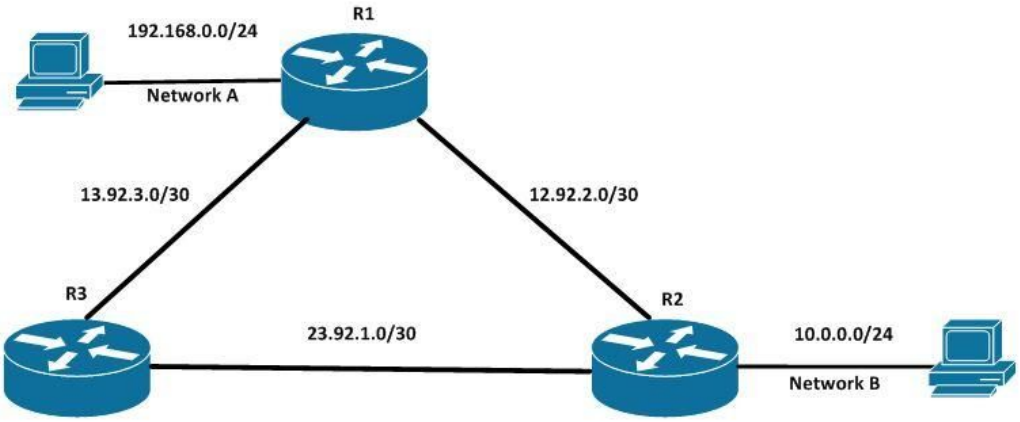


Figura 13.5

Modul de configurare al EIGRP-ului pe un Router Cisco este foarte **asemanator** cu cel al lui RIP sau OSPF. Tot ce trebuie sa facem este sa pornim un proces de EIGRP (identificat prin AS - identic pe toate Routerele) si sa **adaugam retele direct conectate** prin comanda #***network*** (retea + wildcard mask).

NOTA: Unele Router (cu IOS mai vechi de < 15.0, ex: 12.4) necesita comanda #no auto-summary (similara cu cea din RIP)

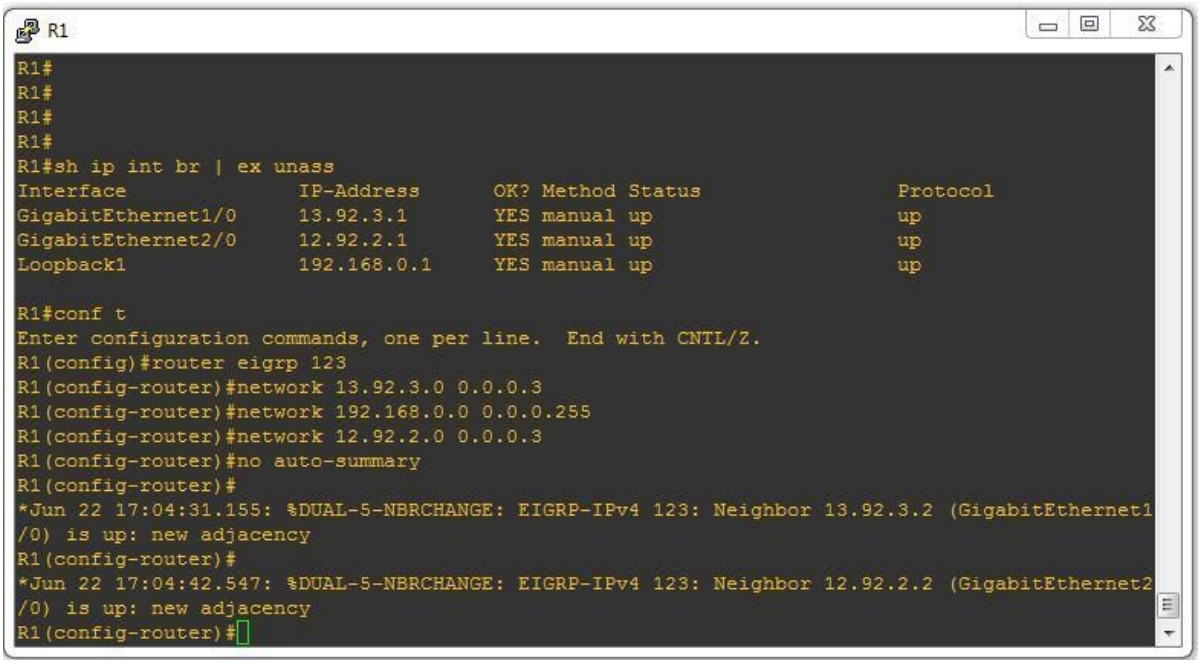


Figura 13.6

```
R1(config)#router eigrp 123
R1(router-config)#network 13.92.2.0 0.0.0.3
R1(router-config)#network 192.168.0.0 0.0.0.255
R1(router-config)#network 12.92.2.0 0.0.0.3
R1(router-config)#no auto-summary
```

ATENTIE: **AS**-ul trebuie sa fie **la fel pe toate Routerele**. Ex: AS 123 -> pe R1, R2, R3 etc. Iata si configul de pe Routerul R3:

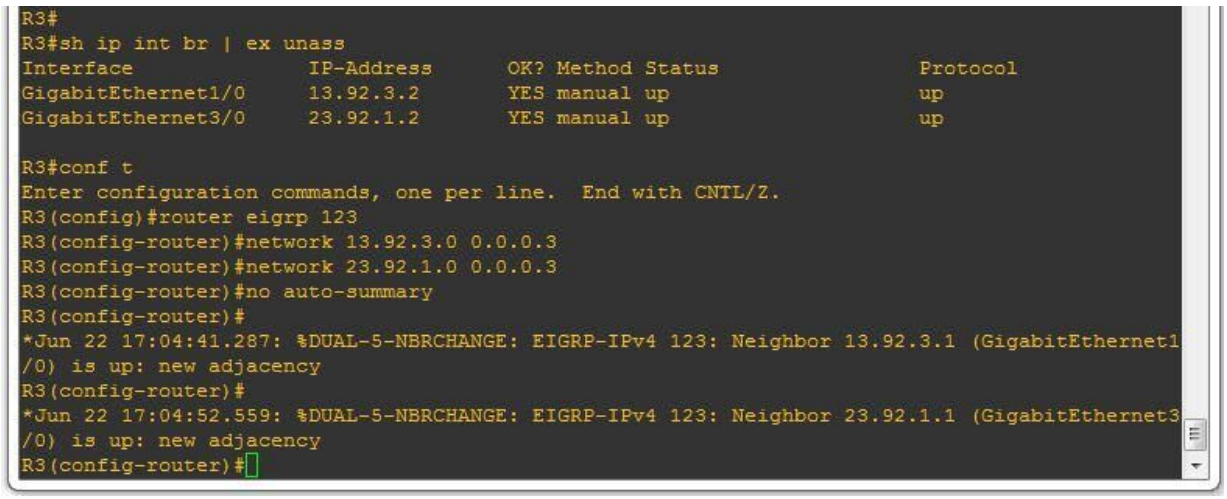


Figura 13.7

Si nu in ultimul rand si pe Routerul 2, aceleasi comenzi de network dar cu alte adrese de retea:

```
R2(config)#router eigrp 123
R2(router-config)#network 10.0.0.0 0.0.0.255
R2(router-config)#network 12.92.2.0 0.0.0.3
R2(router-config)#network 23.92.1.0 0.0.0.3
R2(router-config)#no auto-summary
```

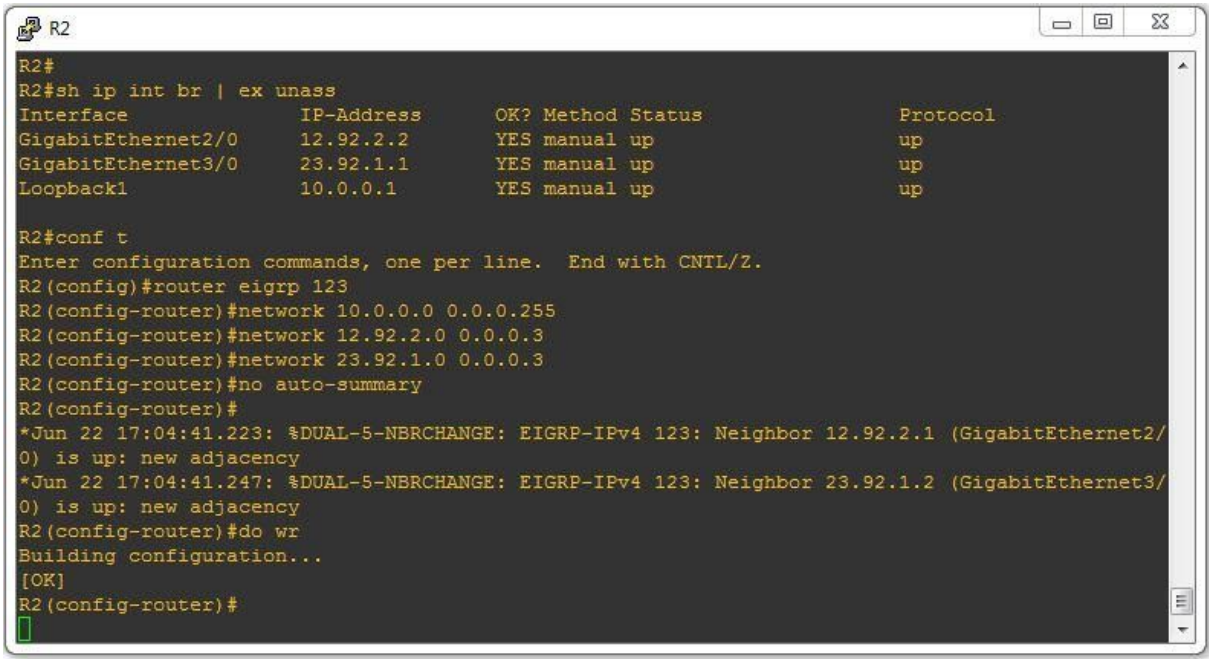


Figura 13.8

Verificarea Configuratiei

Dupa cum poti vedea in fiecare figura de mai sus, odata cu adaugarea retelelor (prin comanda *#network*) **incep sa se formeze adiacentele** (de vecini) dintre Routere. Acum a venit momentul sa verificam si sa vedem cum arata acestea. Iata cateva comenzi utile:

```
R1#show ip eigrp neighbors
```

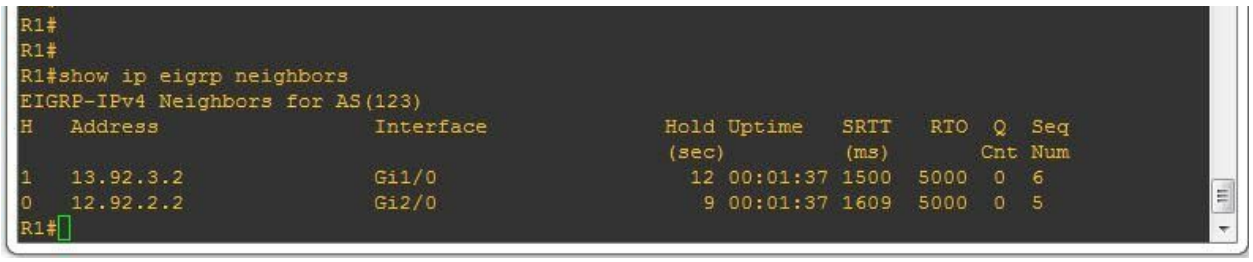


Figura 13.9

Aceasta comanda (#**show ip eigrp neigbors)** ne arata tabela cu Routerele vecine care fac parte din EIGRP. Este un element foarte important (mai ales cand vine vorba de troubleshooting) sa verificam pe fiecare Router in parte faptul ca s-a stabilit o adiacenta EIGRP cu Routerul vecin. Dupa asta putem trece la urmatoarea comanda care ne ajuta sa verificam tabela de rutare.

```
R1#show ip route
```

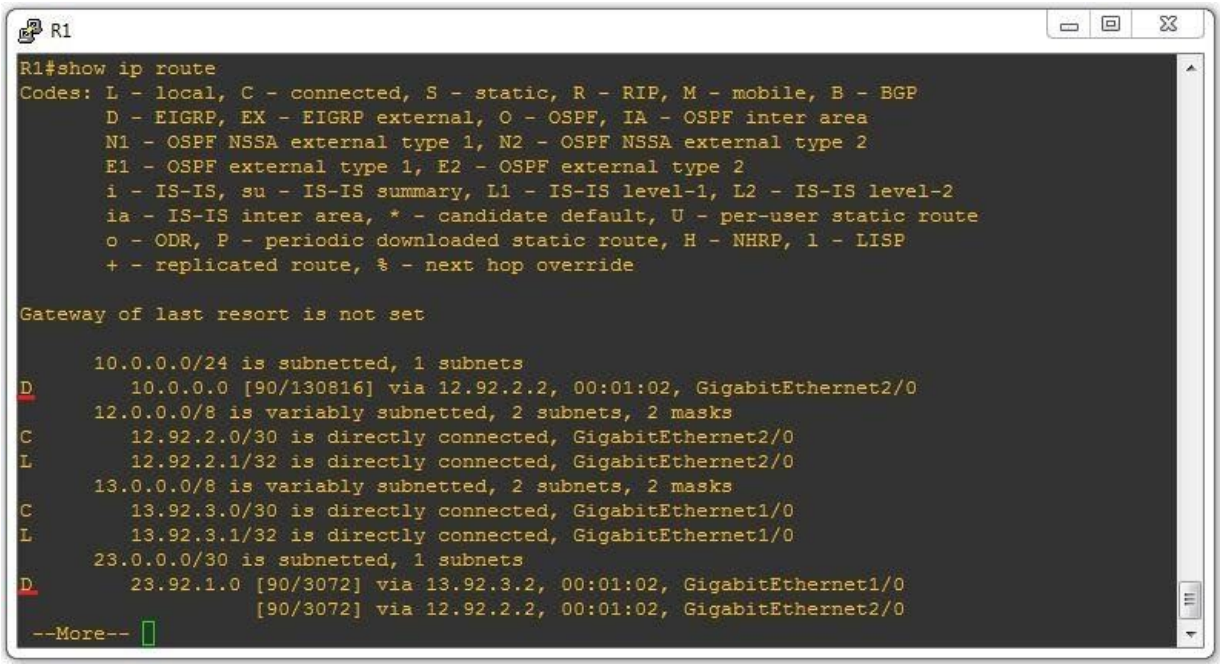


Figura 13.10

Ce ne intereseaza pe noi din acest output sunt **rutele cu D** in fata (care reprezinta **informatie** de rutare, *provenita din protocolul EIGRP*). Dupa cum poti vedea reteaua **10.0.0.0/24** vine de la 12.92.2.2 (un **vecin** al lui R1) si are **metrica** 130816 (aka **FD**). Iata un rezumat:

10.0.0.0 [90 / 130816] via 12.92.2.2

Reteaua AD Metrica (FD) Succesor (Next-Hop)

```
R1#show ip protocols
```

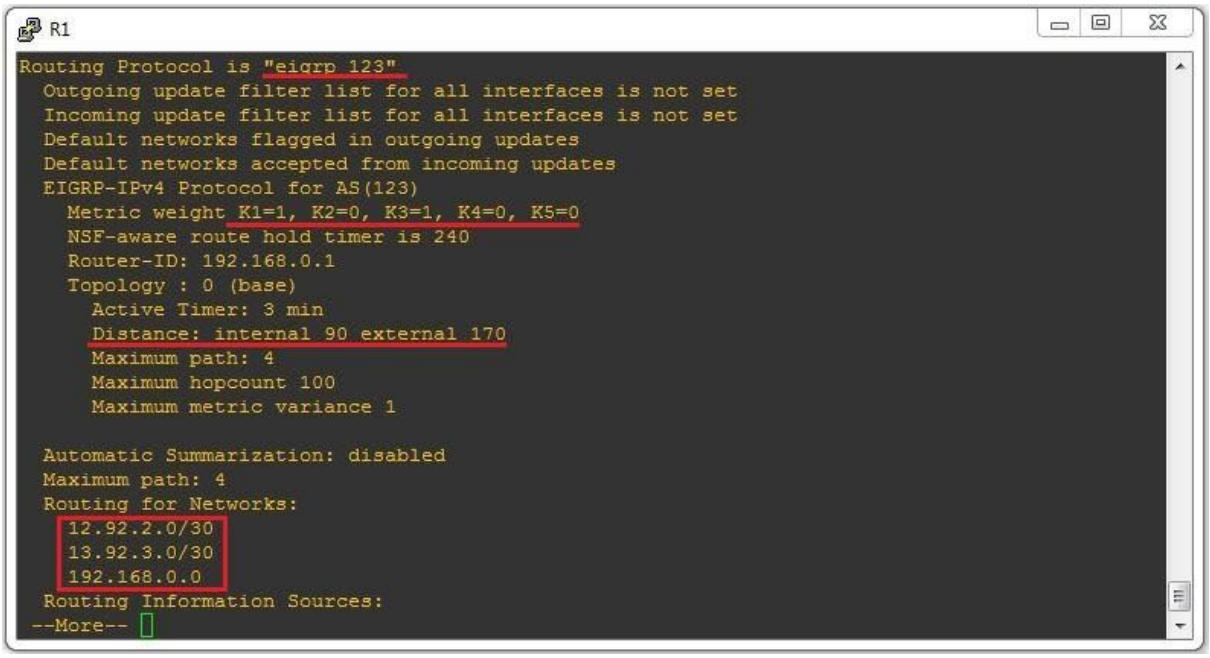


Figura 13.11

In outputul de mai sus poti vedea mai multe detalii despre EIGRP cum ar fi:

- **AS**-ul si valorile **K**
- **Retelele** adaugate in EIGRP
- **RID**-ul, **AD**-ul intern (90) si cel extern (170)

```
R1#show ip eigrp topology
```

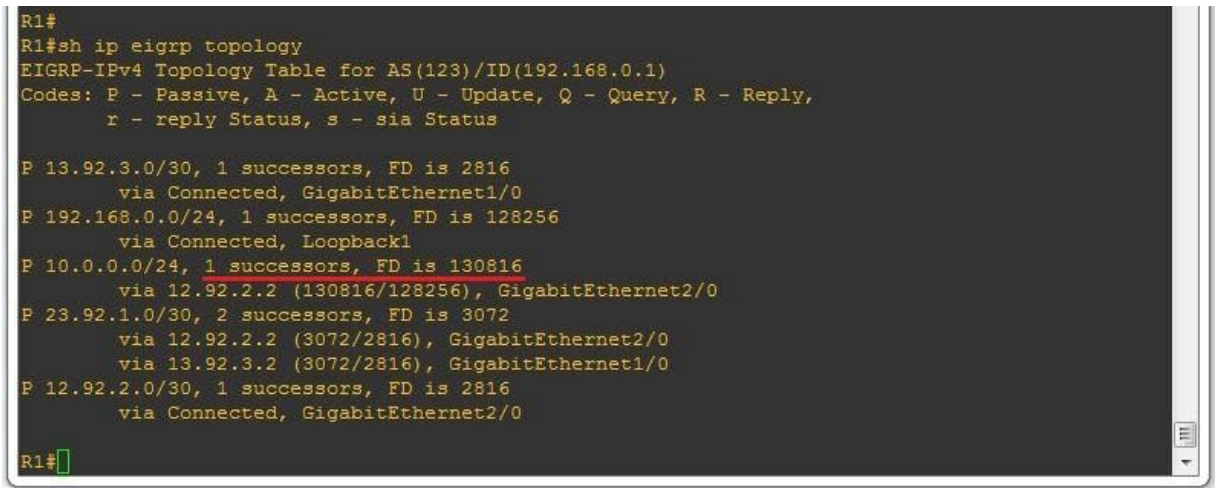


Figura 13.12

```
R1#show run | section eigrp
```

Figura 13.13

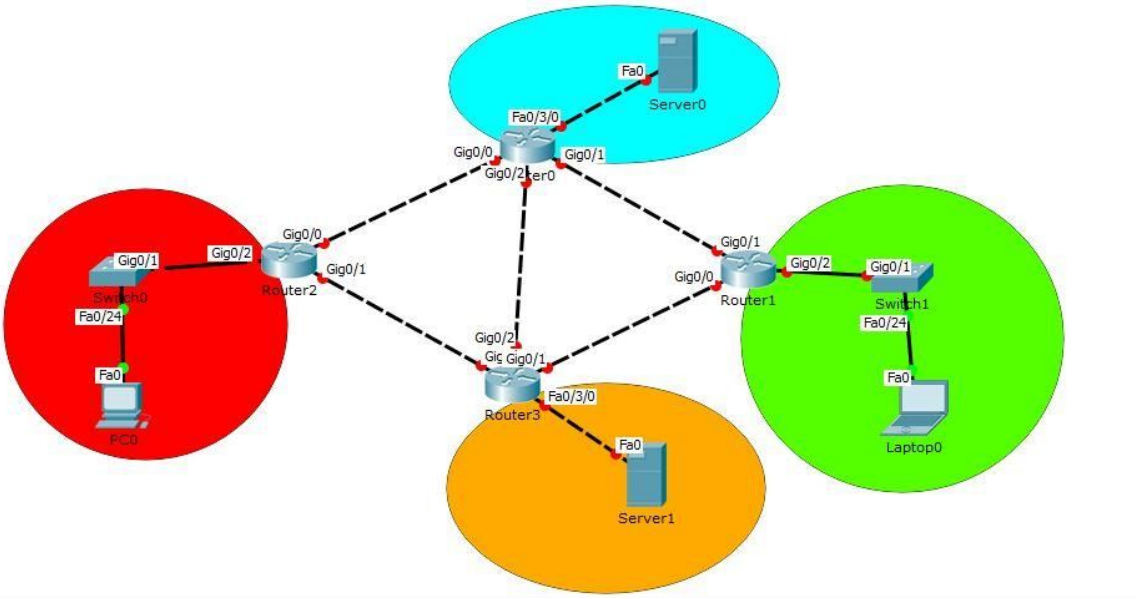
Aceasta comanda ne arata comenzile pe care le-am dat pe R1 pentru configurarea EIGRP-ului. In cazul in care a intervenit o greseala in configurare (un misconfiguration), putem merge in #**router eigrp 123** si aplica aceeasi comanda, dar cu #**no** in fata pentru **stergerea** ei.

Laboratorul #8

Acum am ajuns la partea de laborator (partea practica), pe care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

Acest laborator este similar cu Laboratorul #6 (configurari de baza cu OSPF).

SCOP: Configurari de baza cu EIGRP

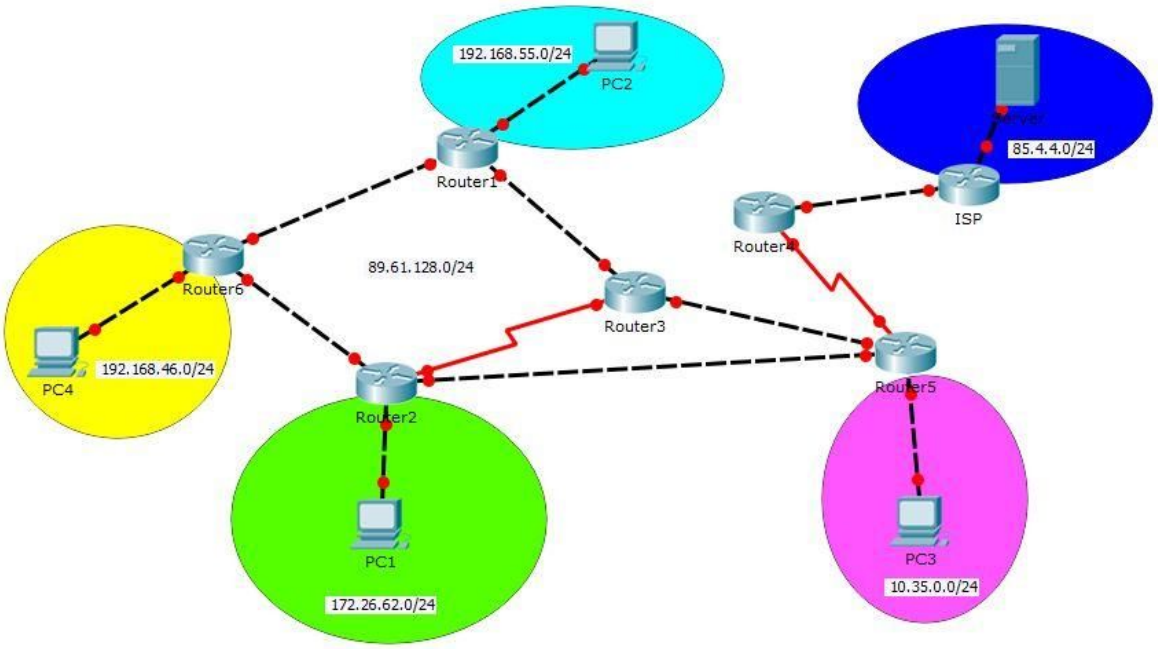


SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Laboratorul #9

Am ajuns la o alta parte de laborator (partea practica), pe care o ai inclusa si in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurarea protocolului de rutare EIGRP intr-o topologie mai complexa



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Capitolul 14 - Concepte de Switching. VLAN-uri si Interfete Trunk & Access

Pentru ca in capitolele anterioare am vorbit mai in detaliu despre Routere si procesul de rutare (practic modul in care se transmit datele prin Internet), in acest capitol vom vorbi mai in detaliu despre Switch-uri. Mai exact vom vedea ce reprezinta conceptul de VLAN, de interfete Trunk si interfete Access pe care marea majoritate a retelelor medii si mari le folosesc.

1) VLAN (Virtual Local Area Network)

VLAN vine de la **Virtual LAN** (Local Area Network) si are rolul de separa (intr-un mod virtual) retelele conectate la acelasi echipament fizic (Switch). **VLAN**-urile ne permit sa separam (din punct de vedere logic) mai multe device-uri (PC-uri, Laptop-uri) conectate la acelasi Switch. Practic pentru a rezuma lucrurile putem face urmatoarea afirmatie:

Un VLAN = O Retea = Un Domeniu de Broadcast

Asadar un VLAN reprezinta o retea. Daca alegem sa creeam 2 VLAN-uri, inseamna ca vom avea 2 retele diferite (asadar si 2 domenii de Broadcast).

VLAN-urile sunt folosite peste tot in marile companii. Da-mi voie sa-ti dau cateva exemple:

#Ex1: Reteaua Wireless **Guest** si **Internal**; provin de la acelasi Router Wireless, dar sunt separate din punct de vedere logic. Asta inseamna ca, by default, nu poti avea acces din reteaua Guest in reteaua Internal si nici invers.

NOTA: Deci astfel obtinem o separare dpvd. logic care duce la **segmentare** si la o crestere a **nivelului** de **securitate** din retea.

#Ex2: in toate companiile medii si mari unde se doreste o separare dispozitivelor pe departamentele companiei (exemplu: departamentul IT (ex: VLAN 45) nu va putea accesa toate resursele din departamentul Marketing (ex: VLAN 91))

NOTA: pentru a identifica diferite departamente, VLAN-urile folosesc un ID (un numar de identificare) unic caruia ii poate fi asociat un nume (pentru identificarea mult mai usoara).

Astfel, ID-ul unui VLAN poate fi in urmatoarele categorii:

- **Standard** VLAN ID – **1 – 1005**
- **Extended** VLAN ID – **1006 – 4094**

Dupa cum poti vedea exista si un range extended de ID-uri care a fost adaugat ulterior, dupa crearea standardului pentru VLAN-uri, scopul acestui range fiind pentru extinderea numarului total de VLAN-uri care pot exista pe un Switch cat si pentru folosirea aplicatilor de rutare pe Switch-uri de nivelul 3.

Inainte de a merge mai departe vreau sa mentionez faptul ca tehnologia VLAN-ul este un standard in industrie si este identificata prin **IEEE 802.1q**. Asta face ca tehnologia sa fie disponibila pe toate echipamentele de retea (a vendorilor precum Juniper, Huawei, HP etc.) si nu doar pe echipamentele Cisco.

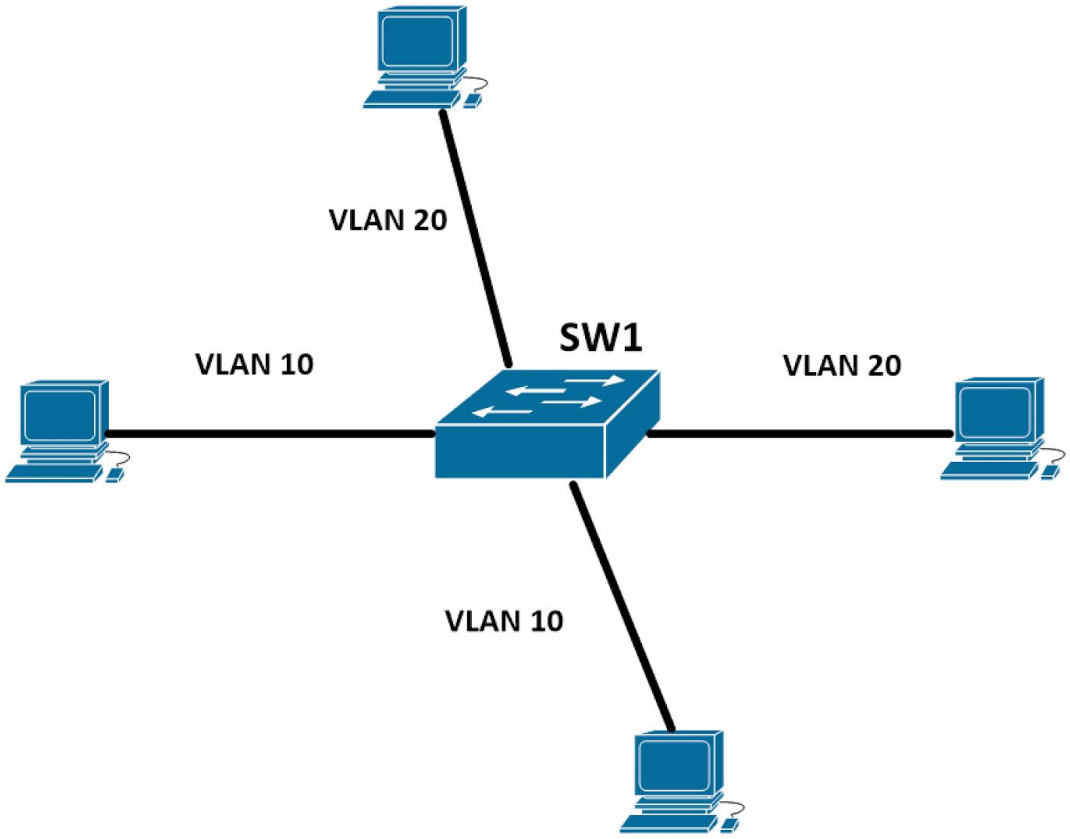


Figura 14.1

lata in figura 14.2 o reprezentare mai clara a PC-urilor care pot comunica intre ele. Astfel, dupa cum poti sa vezi, doar PC-urile din acelasi VLAN pot comunica intre ele.

Toate echipamentele sunt conectate fizic la acelasi Switch, dar dpvd. logic ele sunt separate. De ce? Pentru ca Switch-ul adauga un tag (eticheta) care specific clar faptul ca

doar echipamentele care au acelasi tag (aka. VLAN ID) pot comunica intre ele (in acest scenariu, doar PC-urile din cercul verde, respectiv cele din cercul rosu).

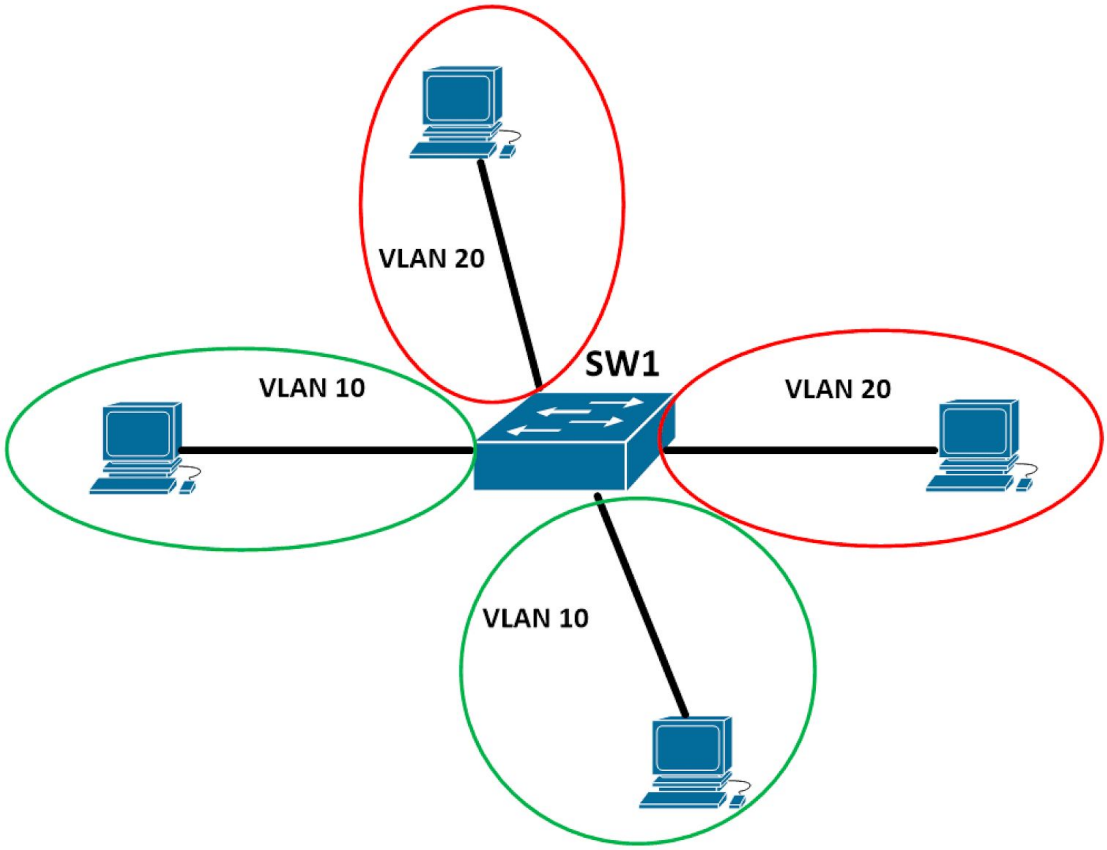


Figura 14.2

Beneficiile VLAN-urilor:

Pentru ca tot vorbim despre VLAN-uri, iata cateva dintre beneficiile pe care acestea le aduc in retele:

1) ***Securitate*** - separarea (izolarea) retelelor la nivel logic

2) ***Design mai bun*** - impartirea unei companii in departamente (fiecare departament reprezentand cate un VLAN (Retea))

3) ***Cresterea Performantei*** - prin limitarea traficului de tip broadcast care creste semnificativ in retele mari

4) ***Scalabilitate*** - se pot adauga foarte usor alte VLAN-uri fara a impacta fluxul retelei

Acestea sunt **reprezentate** printr-un **ID** (un numar de la 1 - 4094).

```
SW1#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0
```

Figura 14.3

Dupa cum poti sa vezi in figura 14.3 de mai sus, pe Switch-urile Cisco exista cateva **VLAN ID-uri rezervate** (nu pot fi utilizate). Aceste ID-uri sunt:

- **1 - VLAN-ul default**, asignat pe toate interfetele Switch-ului (Fa0/1 pana la Fa0/24 in figura de mai sus)
- **1002 - 1005** - aceste VLAN-uri sunt rezervate pentru tehnologii mai vechi (FDDI, Token Ring) care practic, astazi, nu se mai folosesc

Tipuri de interfete pe un Switch

Acum poate te intrebi, avand cele spuse mai devreme, *cum putem aplica aceste VLAN-uri pe Switch? Cum se configureaza ele?*

Inainte de a trece la partea de configurare, trebuie sa mentionez faptul ca, pe Switch, exista 2 tipuri interfete speciale: **Access si Trunk.** VLAN-urile se configureaza pe porturile Switch-urilor, care **trebuie sa fie intr-unul din aceste moduri.**

Astfel, cele 2 denumiri reprezinta:

- **Access** - permite trecerea ***UNUI* singur VLAN**
- **Trunk** - permite trecerea ***mai multor* VLAN-uri**

Porturile de tip **Access** le configuram atunci cand la celalalt **capat se afla un end-device,** (figura 14.4), (PC, Laptop, Server etc.)

Porturile de tip **Trunk** le configuram atunci cand l**a celalalt capat se afla un Switch**, (figura 14.4) deoarece dorim sa permitem trecerea mai multor VLAN-uri prin si intre acestea.

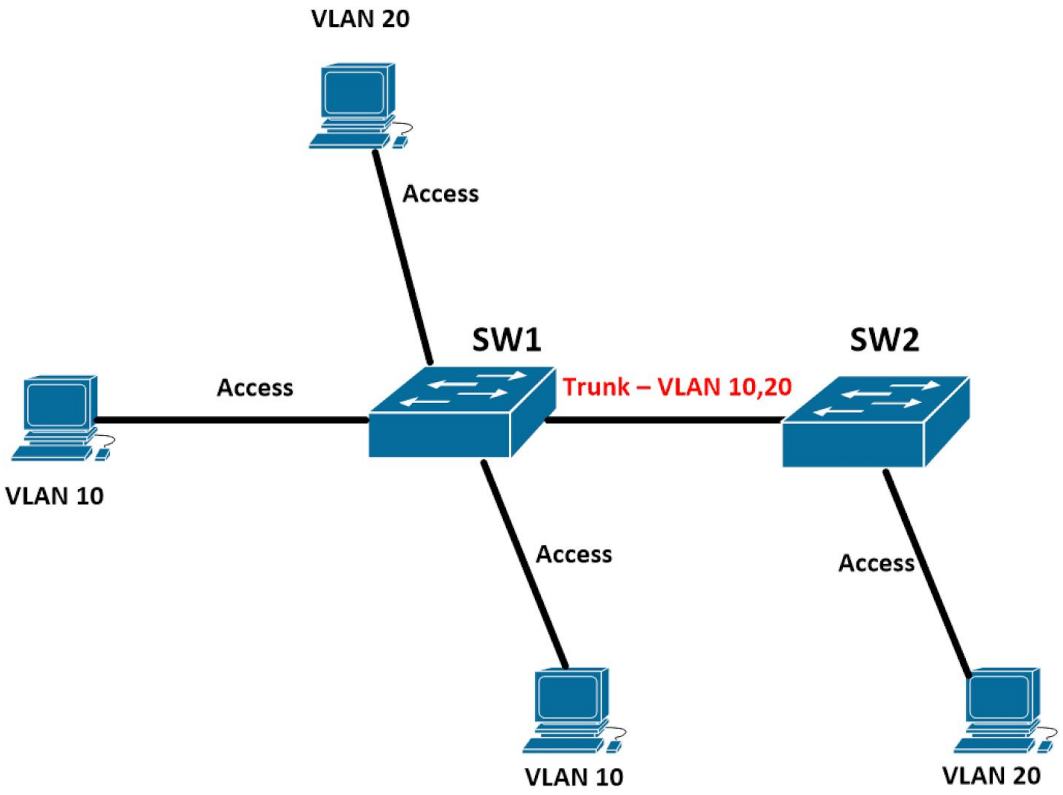


Figura 14.4

Spre exemplu, daca interfata Trunk dintre SW1 si SW2 ar fi avut configurata doar VLAN-ul 10, atunci SW1 nu ar fi putut sa trimita trafic destinat VLAN-ului 20 catre PC-ul conectat la SW2. De aceea este foarte important ca in momentul in care facem setarile unui port in Trunk sa verificam de mai multe ori daca am inclus toate VLAN-urile.

2.1) Configurare VLAN-urilor pe Switch-uri Cisco

Sa presupunem ca avem urmatoarea retea formata din 2 Switch-uri, 4 PC-uri si 2 VLAN-uri (10, respectiv 20):

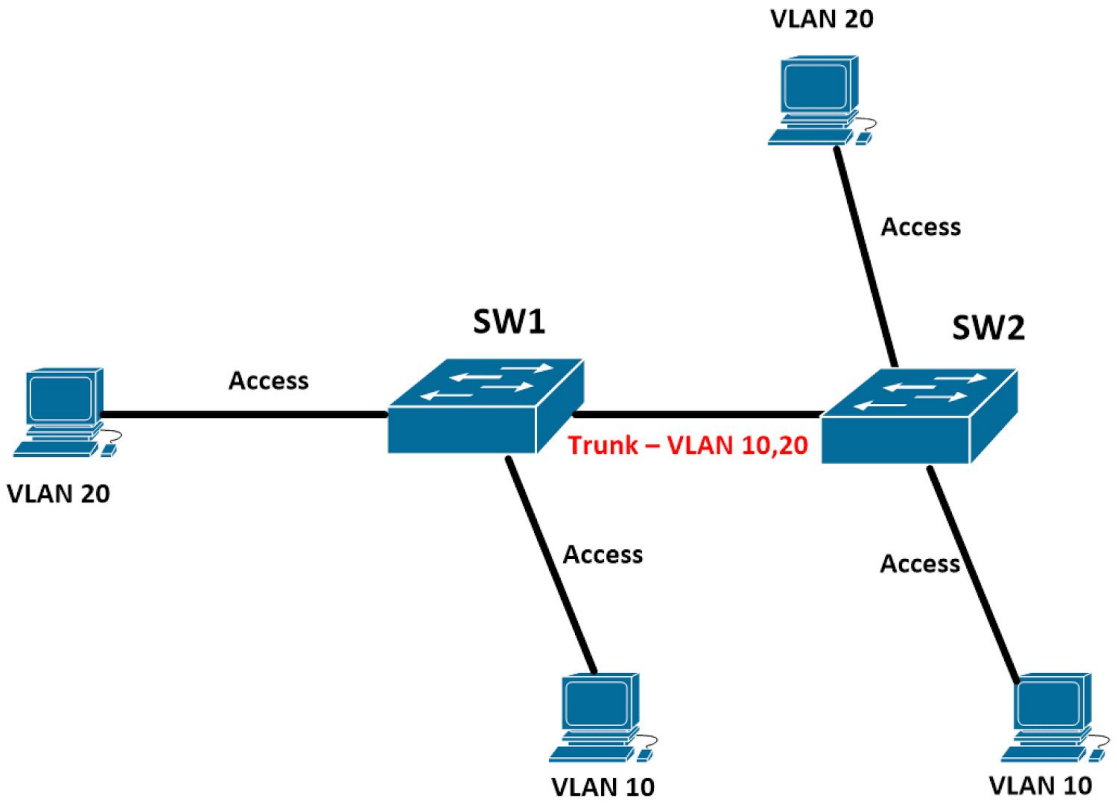


Figura 14.5

2 dintre aceste PC-uri se afla in VLAN-ul 10, iar celelalte 2 se afla in VLAN-ul 20. Pentru a configura un VLAN pe Switch, trebuie sa urmam urmatorii pasi (din config terminal):

```
SW1(config)#vlan 10
```

```
SW1(config-vlan)#name IT
```

```
SW1(config)#vlan 20
```

```
SW1(config-vlan)#name HR
```

In acest stadiu, avem VLAN-urile create si le-am asignat un nume (IT pentru VLAN-ul 10 si HR pentru VLAN-ul 20). Mergand mai departe trebuie sa setam aceste VLAN-uri pe interfetele dispozitivelor conectate la Switch.

NOTA: aceste VLAN-uri se configureaza pe **FIECARE** Switch din retea (SW1 si SW2)

2.2) Configurare Interfata Access

O ***interfata Access*** permite trecerea traficului dintr-un SINGUR VLAN (in cazul nostru 10 **sau** 20) prin ea. Aceasta va fi *configurata pe legatura dintre Switchuri si PC-uri* in urmatorul mod:

a) Vom configura interfata FastEthernet 0/1 ca fiind Access pentru VLAN-ul 10:

```
SW1(config)#interface fa0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
```

b) Iar interfata FastEthernet 0/2 o vom seta pentru VLAN-ul 20, tot Access:

```
SW1(config)#interface fa0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 20
```

2.3) Configurare Interfata Trunk

O ***interfata Trunk*** permite trecerea traficului din mai multe VLAN-uri (in cazul nostru 10 **si** 20) prin ea. Aceasta va fi *configurata pe legatura dintre Switchuri* in urmatorul mod:

```
SW1(config)#interface fa0/24
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,20
```

Pana in acest moment am configurat pe cele 2 echipamente de retea, VLAN-urile 10 si 20, am setat porturile la care sunt conectate calculatoarele, in modul Access, am adaugat fiecare PC in VLAN-ul sau, iar la sfarsit am configurat interfata dintre Switch ca fiind Trunk.

Acum, PC-urile din acelasi VLAN (10 - 10) **vor putea comunica**, iar cele aflate in VLAN-uri diferite **nu vor putea comunica** (10 - 20).

ATENTIE: aceste configurari trebuie aplicate si pe celelalte Switch-uri, in acest caz si pe SW2

2.4) Verificarea Setarilor

Acum, dupa atata configurat este timpul sa si verificam ceea ce am setat pe Switch-urile noastre. Elementul #1 care ne intereseaza este, bineinteles, conectivitatea. Dar, conectitivatea intre care end-device-uri? Intre toate? Ei bine, nu... De ce? Pentru ca noi avem end-device-uri in 2 VLAN-uri diferite, iar aceste nu pot comunica intre ele (doar device-urile care apartin aceluiasi VLAN).

Deci, vom da ping pentru echipamentele din VLAN 10, apoi pentru echipamentele din VLAN 20, iar in cele din urma (pentru a demonstra cele spuse anterior), vom da ping intre 2 device-uri din VLAN-uri diferite.

Verificarea pentru VLAN 10:

`>ping 192.168.10.10` (sau adresa IP a PC-ul din VLAN-ul 10)

Verificarea pentru VLAN 20:

`>ping 192.168.20.20` (sau adresa IP a PC-ul din VLAN-ul 20)

Verificarea pentru VLAN 10 - 20 (verificarea se face de pe PC-ul din VLAN-ul 20):

`>ping 192.168.10.10` (sau adresa IP a PC-ul din VLAN-ul 20)

Daca intampinam anumite probleme, iar pingul nu functioneaza putem verifica pe Switch cu urmatoarele comenzi:

```
SW1#show vlan
```

```
SW1#show interfaces trunk
```

```
SW1#show run
```

In urma acestor comenzi putem vedea daca avem sau nu create VLAN-urile si pe ce porturi sunt ele alocate.

3) Rutarea intre VLAN-uri

In aceasta sectiune vom discuta despre modurile in care putem face **rutarea intre VLAN-uri**. By default, **Switch-urile nu stiu sa lucreze cu pachete** IP (doar cu adrese MAC), ceea ce inseamna ca acestea **nu** vor putea face transfer de date (rutare) dintr-un VLAN in celalalt. Datorita acestui fapt, avem nevoie de procesul de Rutare intre VLAN. Acesta poate avea loc in 3 moduri:

A) Folosind un Router cu atatea interfete cate VLAN-uri exista (acest model nu este unul scalabil, deoarece pe Routere avem un numar limitat de interfete)

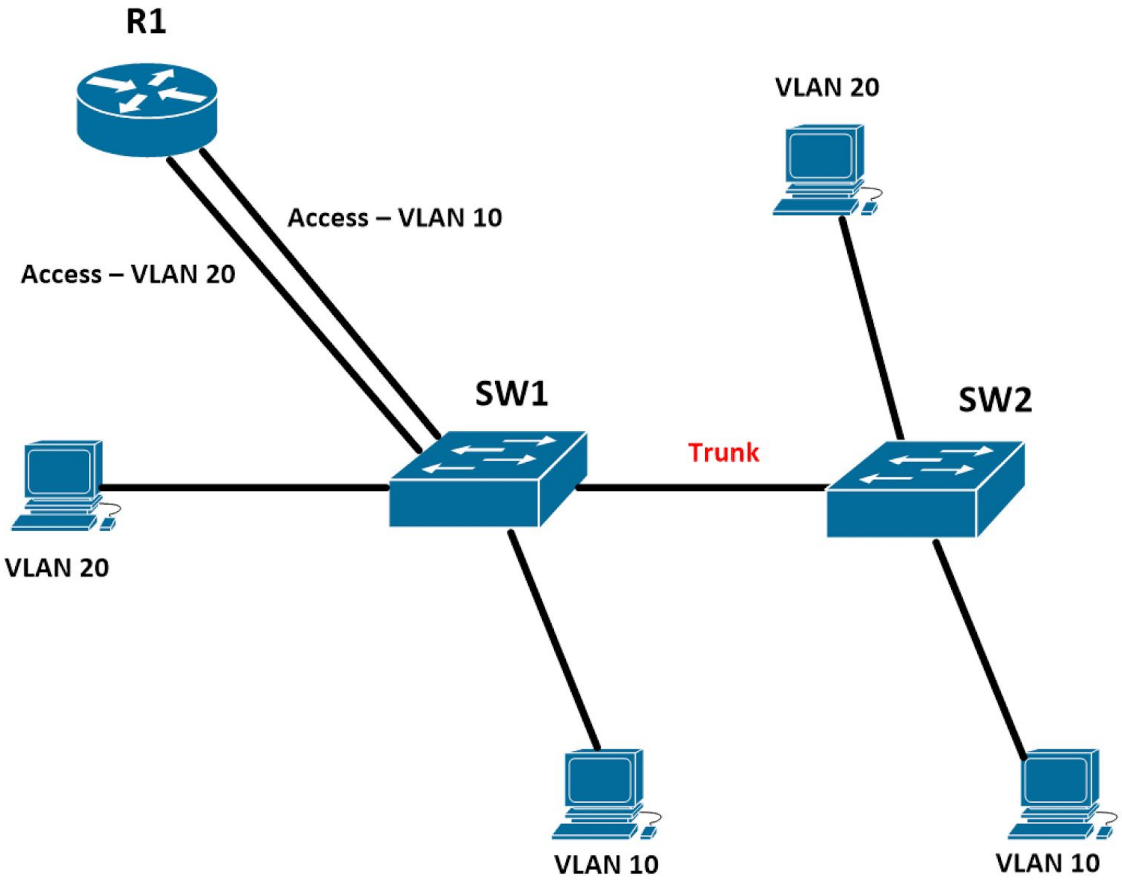


Figura 14.9

Acesta este probabil cel mai ineficient mod de a face rutarea intre VLAN-uri. De ce? Pentru ca avem nevoie de atatea interfete pe Router, cate VLAN-uri exista (ceea ce, in 99% din cazuri, nu este posibil pentru ca Routerul, by design, are un numar mic de interfete 3 - 5). Asadar, aceasta metoda nu scaleaza si avem nevoie de alte optiuni.

B) Folosind un Router conectat la Switch cu o singura interfata (Trunk) prin care vom permite trecerea mai multor VLAN-uri - acest model se numeste **Router-on-a-Stick (RoaS)**

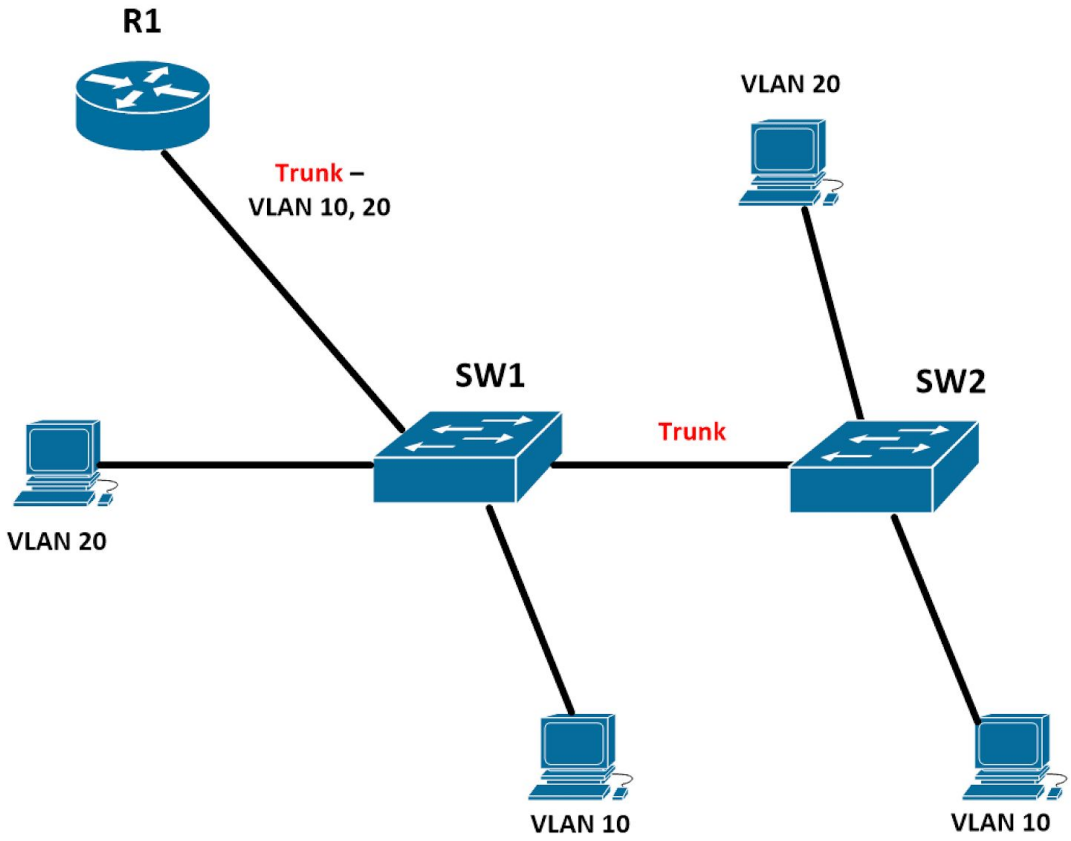


Figura 14.10

Dupa cum am spus si mai devreme, acest mod de rutare a traficului intre VLAN-uri se numeste Routing on a Stick (RoaS) si va fi subiectul discutiei noastre. Acest mod de a face rutarea este unul mai invechit, dar destul de des folosit in retele. Din experienta iti pot spune faptul ca, chiar si marile corporatii folosesc acest model pentru ca este simplu de implementat, iar costurile sunt mult reduse.

Conceptul de **Router-on-a-Stick** este unul foarte simplu. Acesta permite trecerea mai multor VLAN-uri printr-o singura interfata (Trunk) conectata intre Router si Switch. Pe Switch, interfata va fi configurata ca fiind Trunk, iar pe Router se vor creea **subinterfete** pentru fiecare VLAN in parte.

Trebuie in schimb avut grija la nivelul de incarcare a benzii de retea (bandwidth) pentru ca, pana la urma 2 sau mai multe VLAN-uri folosesc ACEEASI interfata pentru a comunica intre ele. In acest caz se recomanda ca interfetele Routerului sa fie cel putin de 1 Gbps.

C) Folosind un Switch de Nivelul 3 (L3) - capabil sa faca rutarea (sa trimita pachete intre VLAN-uri)

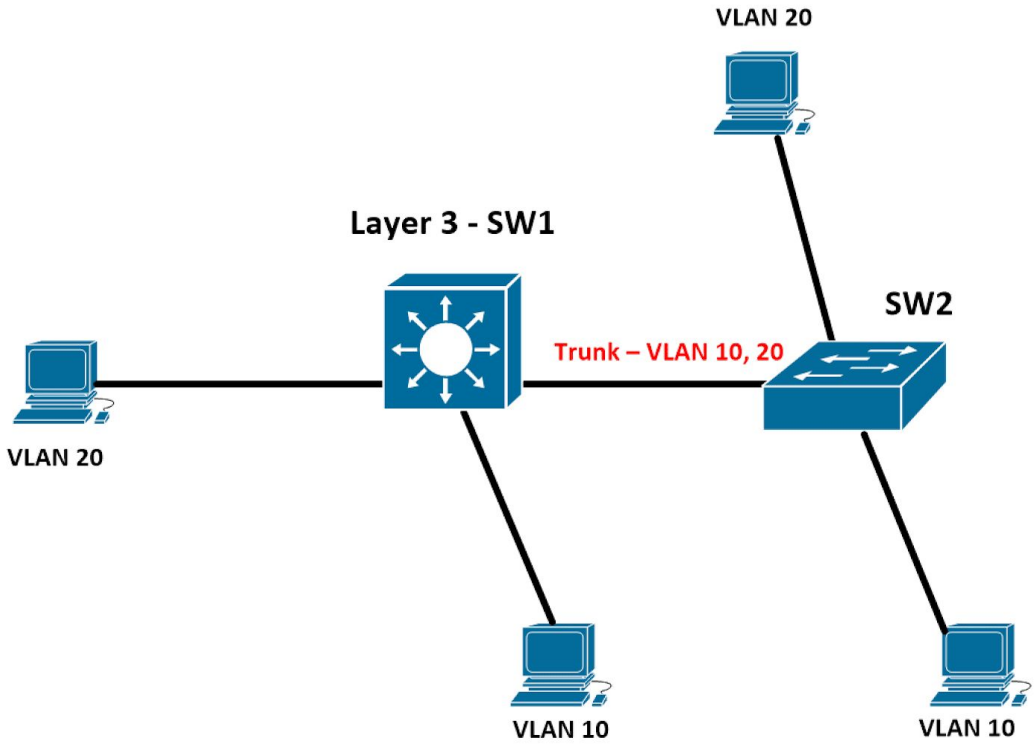


Figura 14.11

Acest mod de a face rutarea intre VLAN-uri este, de departe, cel mai rapid si eficient. De ce? Pentru ca avem la mijloc un Switch Layer 3 care este capabil sa faca rutarea intre VLAN-uri cu mare usurinta. Avantajul folosirii unui astfel de Switch se datoreaza faptului ca el ne ofere viteze mult mai mari (in reteaua locala) pentru ca rutarea se face in Hardware si nu in Software (asa cum o fac Routerele). Switch-urile layer 3 pot fi o alternativa foarte buna de a face rutarea intre VLAN-uri, doar ca este important sa retinem faptul ca aceste “vin la pachet” cu un cost (pret) mai mare. Despre acest subiect vom vorbi mai multe in urmatoarea carte :)

Configurare Rutarea intre VLAN-uri (Router-on-a-Stick)

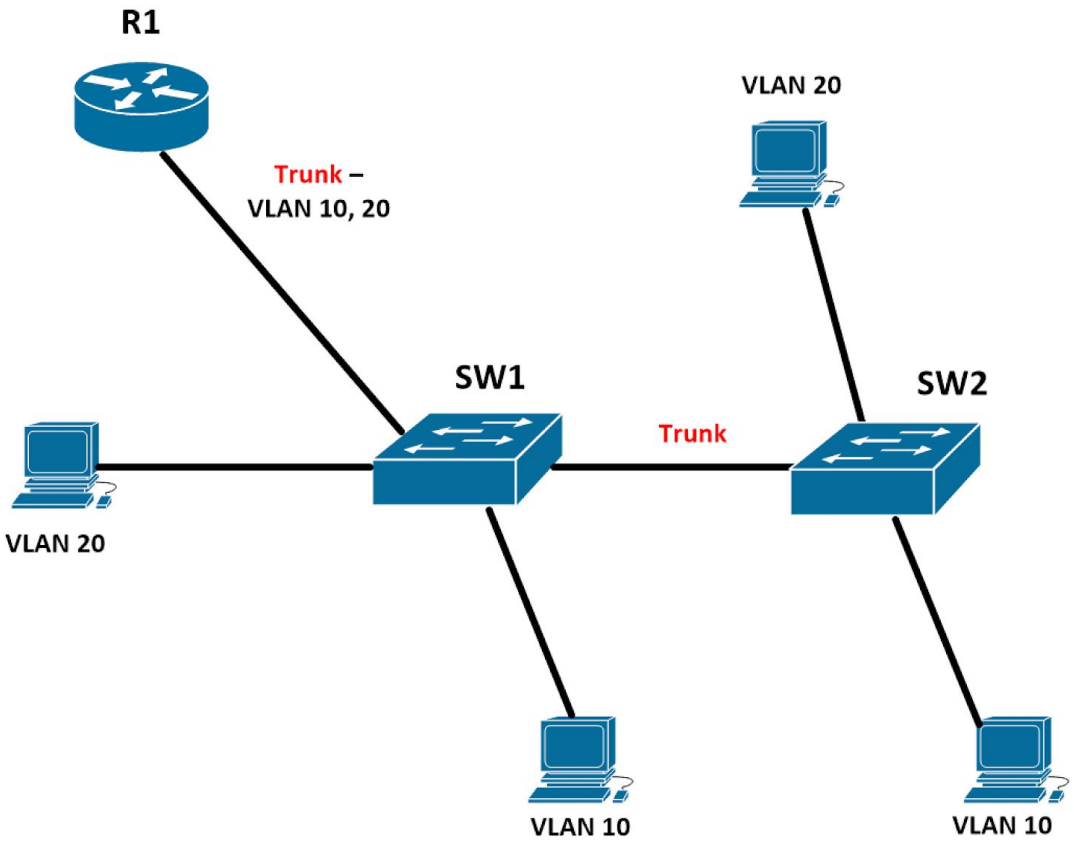


Figura 14.12

* Scenariu:

Sa luam scenariul de mai sus. Presupunem ca, recent am creat o companie, pe nume **RoaS**, care creste foarte repede. La inceput eram organizati intr-un singur departament, dar cum cresterea este foarte rapida, acest design nu mai este unul scalabil si securizat; asa ca ne-am hotarat sa impartim compania in *mai multe departamente* (IT si Vanzari , deocamdata).

Fiecare dintre aceste departamente va avea **un spatiu de adresare propriu** (/24) cu IP-urile `192.168.10.0/24` - IT si `192.168.20.0/24` - Vanzari.

Scopul nostru este ca aceste 2 departamente sa poate comunica intre ele. Implicit, ele nu pot vorbi unul cu celalalt deoarece este necesara existenta unui echipament de nivelul 3 (Router sau Switch de Nivelul 3).

Un alt factor, pe care compania noastra trebuie sa il aiba in vedere este **costul** acestor echipamente. Momentam avem buget doar pentru un Router, deoarece un Switch de Nivelul 3 este mult mai scump.

Asadar, pe langa cost, fiind limitati si de numarul de interfete disponibile pe Router (in general 2-3), solutia pe care o avem la dispozitie pentru a configura **Rutarea intre VLAN-uri** este Router-on-a-Stick.

Pentru a configura Router-on-a-Stick, trebuie sa setam link-ul care conecteaza Switch-ul la Router, ca fiind Trunk; pe Router trebuie sa pornim interfata fizica si sa configuram atatea subinterfete (ex: Fa0/0.10) cate VLAN-uri avem:

```
SW1(config)#interface Fa0/24
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,20

R1(config)#interface Fa0/0
R1(config-if)#no shutdown
```

Acum este momentul sa configuram subinterfetele pentru fiecare VLAN in parte. In scenariul de mai sus, avem 2 VLAN-uri, 10 si 20. Pentru a configura aceste subinterfete pentru fiecare VLAN in parte, trebuie urmati pasii de mai jos:

```
R1(config)#interface Fa0/0.10
R1(config-if)#encapsulation dot1q 10
R1(config-if)#ip address 192.168.10.1 255.255.255.0
R1(config)#interface Fa0/0.20
R1(config-if)#encapsulation dot1q 20
R1(config-if)#ip address 192.168.20.1 255.255.255.0
```

Astfel, aplicand aceste comenzi pe Router va fi posibila comunicarea intre retele.

Verificarea Setarilor RoaS

Pentru a verifica ceea ce am configura mai devreme, pe Router, vom da urmatoarele comenzi:

```
R1#show ip interface brief
```

```
R1#show run
```
//si cautam interfata pe care am configura subinterfetele

De asemenea, pe Switch trebuie sa verificam daca interfata intre el si R1 este intradevar trunk prin comanda:

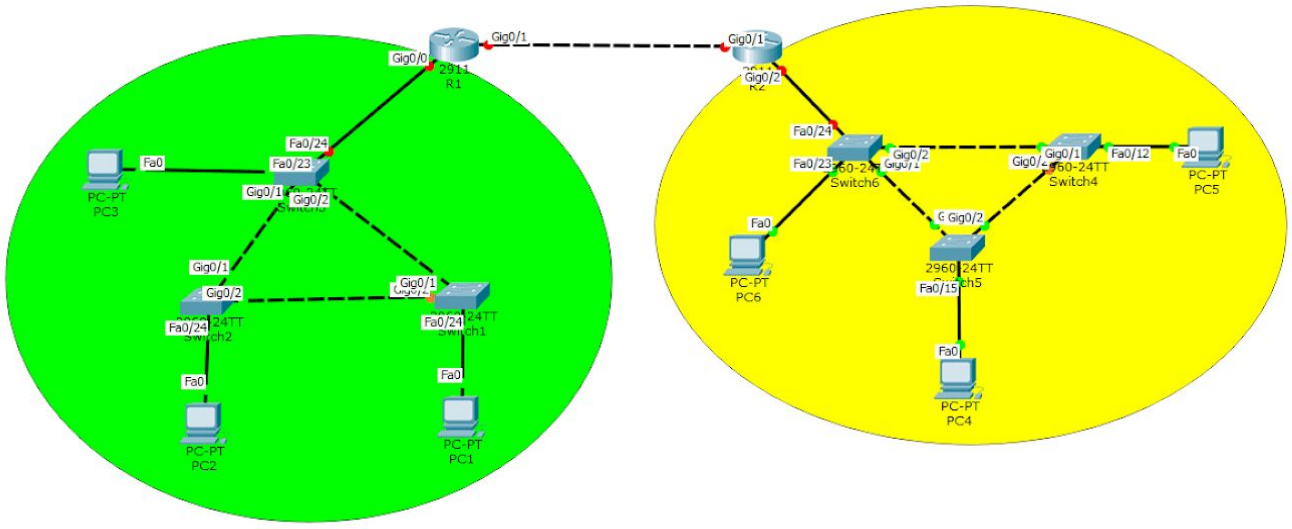
```
SW1#show interfaces trunk
```

```
SW1#show run
```

Laboratorul #10

Acum am ajuns la partea de laborator (partea practica), care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aprofundarea cunostintelor de Switching (VLAN-uri, Trunk-uri si Rutare intre VLAN-uri)



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Capitolul 15 - Servicii de Retea (DHCP, ACL, NAT)

In urmatoarea sectiune vom vorbi despre cateva servicii de retea (DHCP si NAT) si un mod prin care ne putem proteja reteaua. Vom folosi ca referinta, topologia de mai jos:

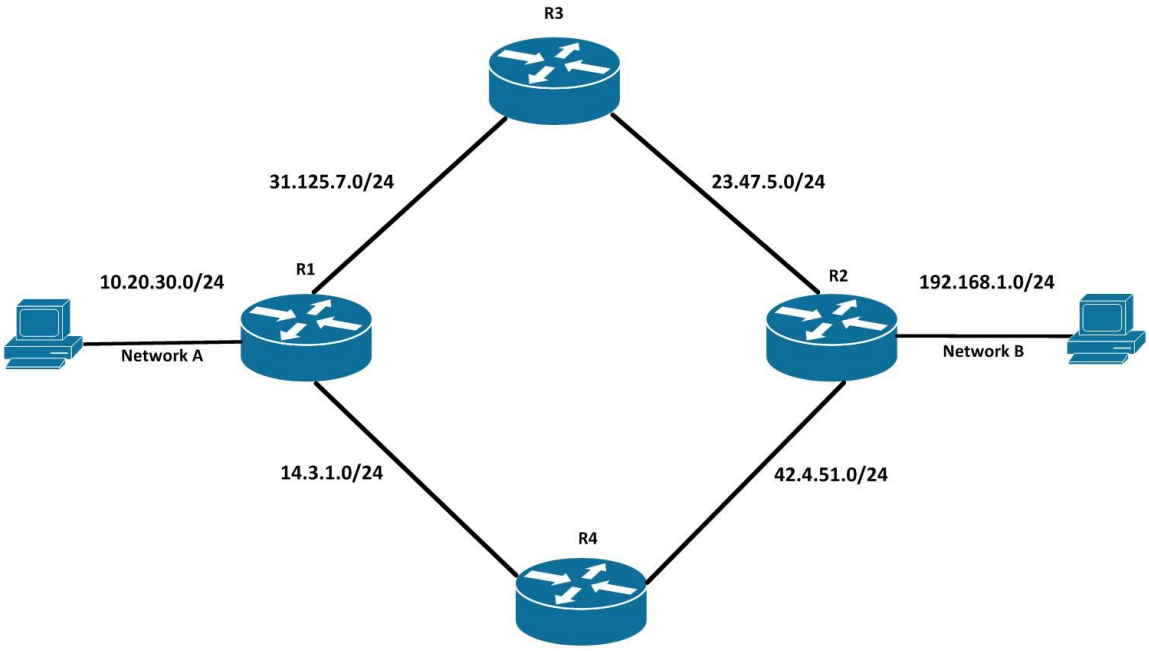


Figura 15.1

1) DHCP

DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) este un protocol de retea care ne ofera in **mod dinamic** urmatoarele informatii:

1) *Adresa IP + Masca*

2) *Default Gateway*

3) *DNS Server*

Adresa IP si masca de retea ne vor ajuta sa identificam fiecare dispozitiv din retea (IP-ul) si sa stabilim dimensiunea retelei (masca de retea).

Serverul DNS ne ajuta cu **“translatarea” numelui** (ex: google.ro) intr-o adresa IP (216.58.214.227) Toate aceste informații sunt furnizate de un server (in retelele mai mici, de asta se ocupa Routerul).

Cum functioneaza DHCP ?

In momentul in care un dispozitiv (PC, smartphone, tableta, SmartTV etc.) se conectează la retea va trimite o cerere Broadcast (catre toate dispozitivile din retea) in speranta ca va gasi un server care sa-i aloce o adresa IP :

1) DHCP Discover

In momentul in care un server DHCP (in retele mici va fi in general un Router Wi-Fi) vede un astfel de mesaj in retea, va raspunde imediat cu un:

2) DHCP Offer (care conține informațiile enumerate mai sus)

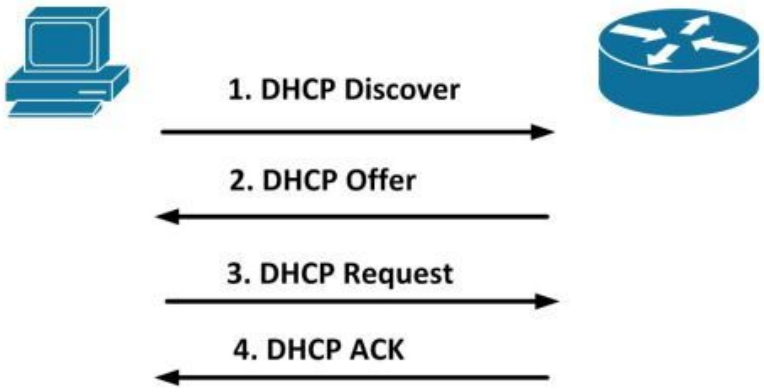


Figura 15.2

In final, dispozitivul (PC-ul in acest caz) va fi de acord cu “oferta” propusa de server DHCP si va trimite o cerere pentru aceasta:

3) DHCP Request

Dupa ce primeste acest mesaj serverul DHCP va raspunde cu un

4) DHCP ACK *// in semn de confirmare a cererii primite de la dispozitiv (PC)*

Configurare DHCP pe Router

Pentru a configura un Router ca DHCP Server trebuie sa avem in vedere urmatoarele elemente:

1) Adresa de retea si masca - (10.20.30.0/24)
2) Adresa IP a gateway-ului (router-ului) - (10.20.30.1)
3) Adresa IP a serverului DNS - (8.8.8.8)

Fiecare din aceste 3 elemente sunt esentiale pentru asigurarea conectivitatii la Internet la orice end-device (PC, telefon, server etc). Iata si cum le putem configura:

```
R1(config)#ip dhcp excluded-address IP_1 IP_2
R1(config)#ip dhcp pool NUME
R1(config-dhcp)#network 10.20.30.0/24
R1(config-dhcp)#default-router 10.20.30.1
R1(config-dhcp)#dns-server 8.8.8.8
```

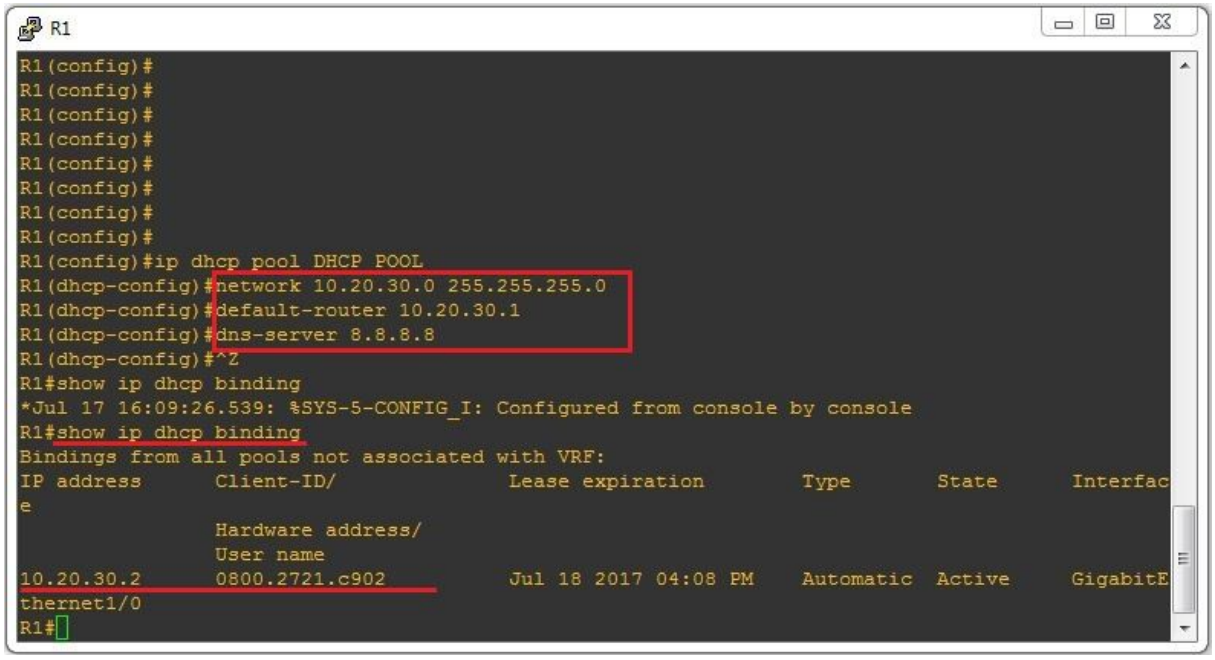


Figura 15.3

Iata si cateva comenzi de verificare:

```
R1#show ip dhcp binding
R1#show run | dhcp
```

Iata-ne si pe Ubuntu (Linux) alocand o adresa IP dinamica de la serverul DHCP creat mai devreme. Folosim #**dhclient** enp0s3 (care reprezinta numele interfetei) si comanda #**ifconfig**, respectiv ping, pentru a verifica adresa IP si conectivitatea cu routerul R1.

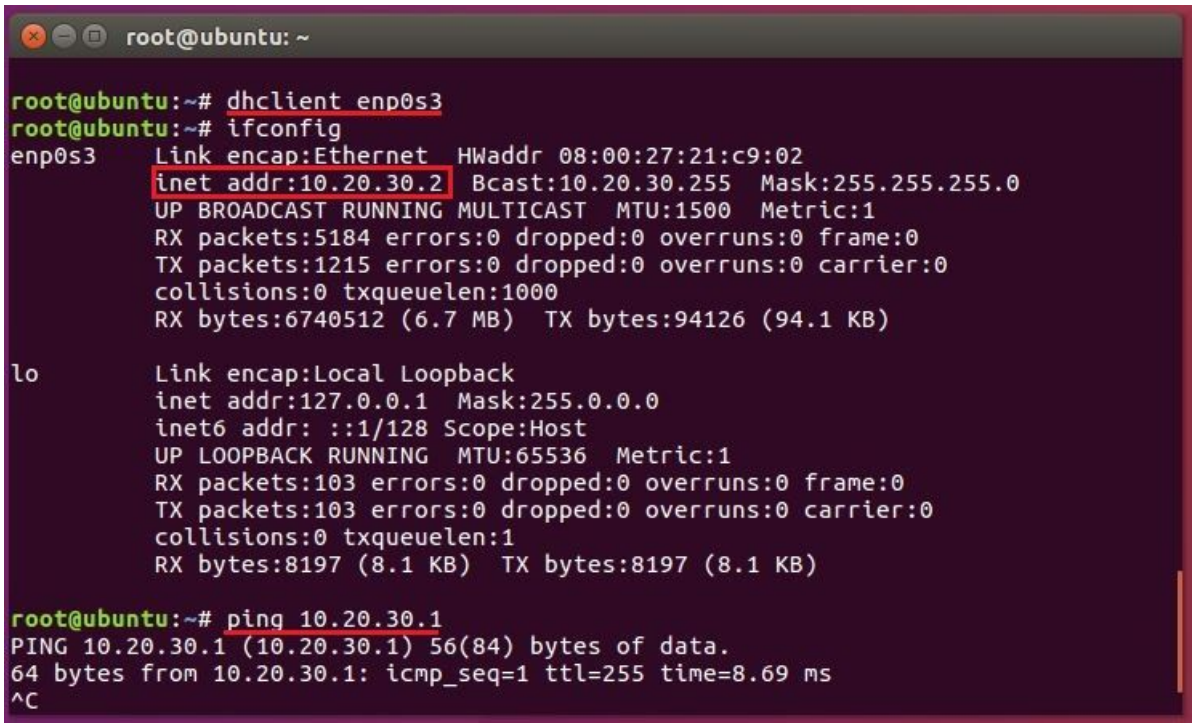


Figura 15.4

Configurare adresa IP dinamica (prin DHCP) pe Windows

Acum vom lua un alt exemplu si vom vedea modul in care putem aloca o adresa IP prin DHCP pe Windows XP / 7 / 8 / 10. Vom vedea cum putem face asta atat din linie de comanda (CMD) cat si din modul grafic.

1) DHCP din CMD

Modul in care putem aloca o adresa IP prin DHCP in Windows, din CMD, este prin comanda **>ipconfig /renew**

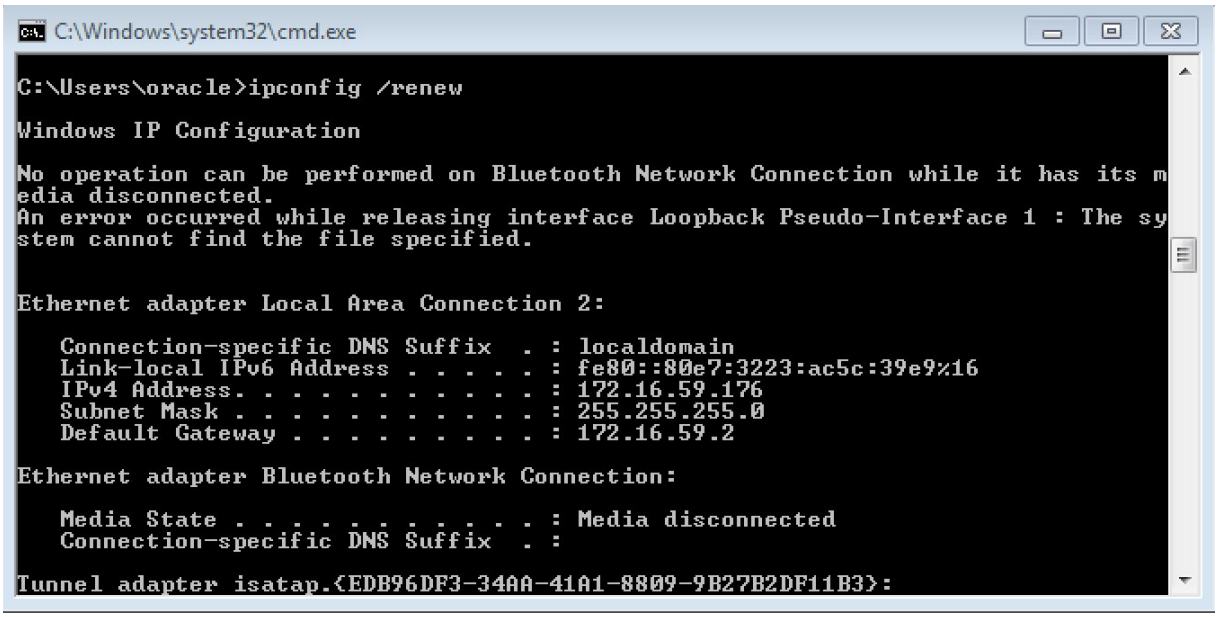


Figura 15.5

Rezultatul ? O adresa IP, masca si default gateway-ul. Daca dorim sa aflam mai multe informatii (precum DNS-ul, adresa MAC sau chiar serverul DHCP) avem la dispozitie comanda:

>ipconfig /all

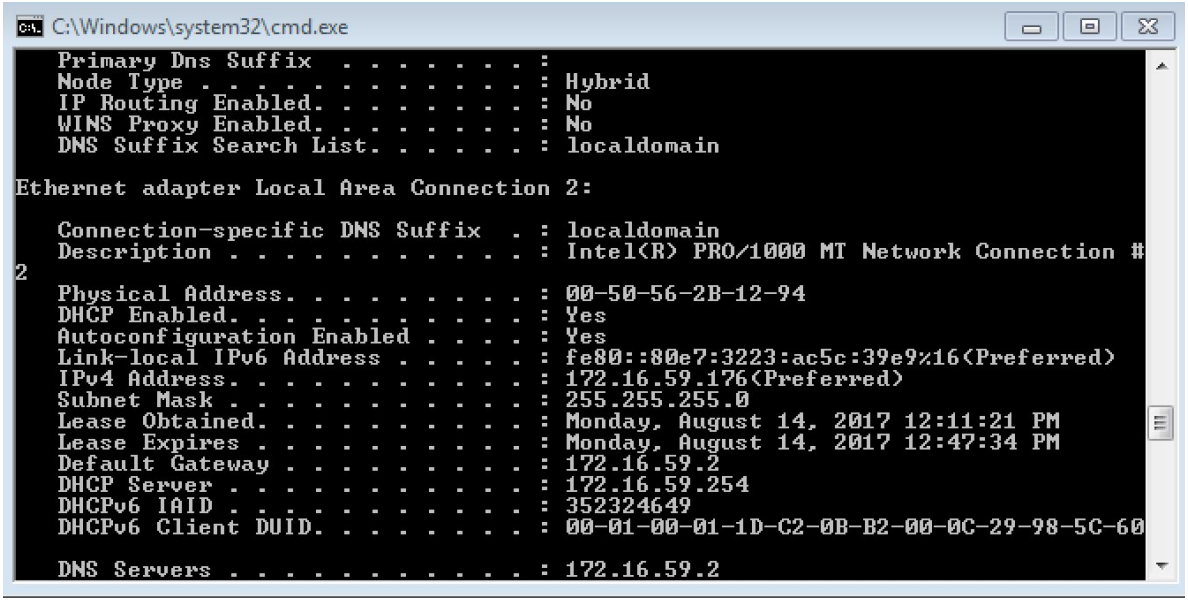


Figura 15.6

2) DHCP prin Modul Grafic

Odata ajunsi in acest punct in care putem configura o adresa IP:

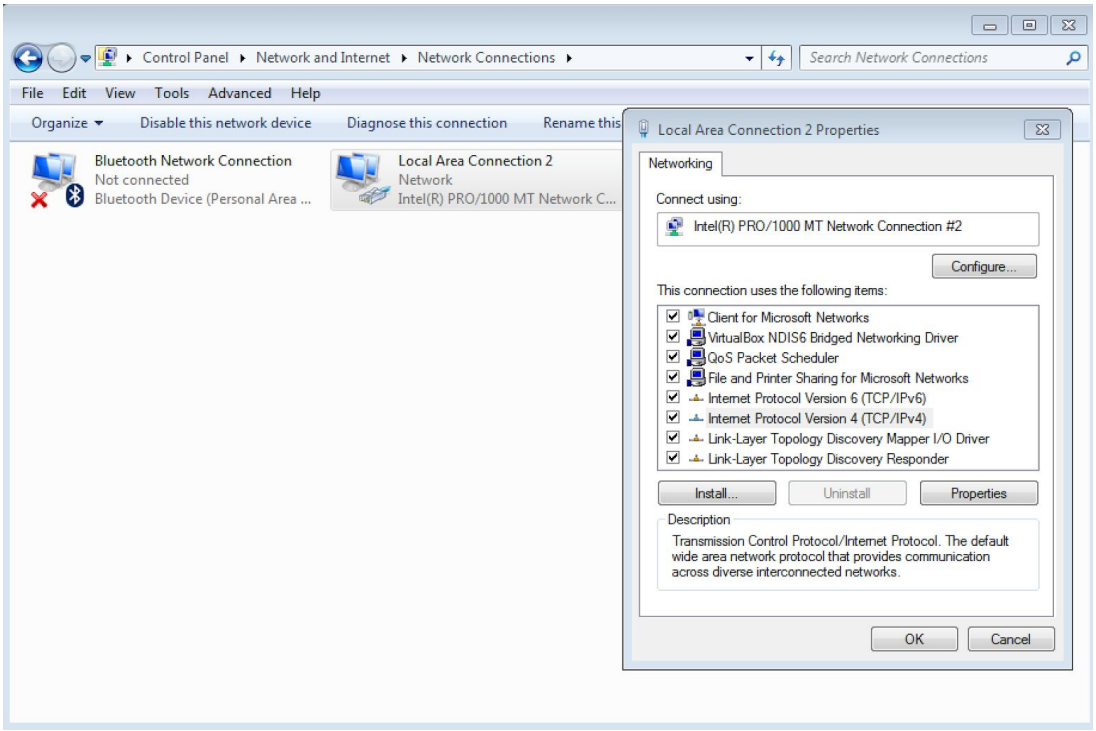


Figura 15.7

Vom selecta IPv4 -> “**Properties**” si avom ajunge la fereastra din figura de mai jos:

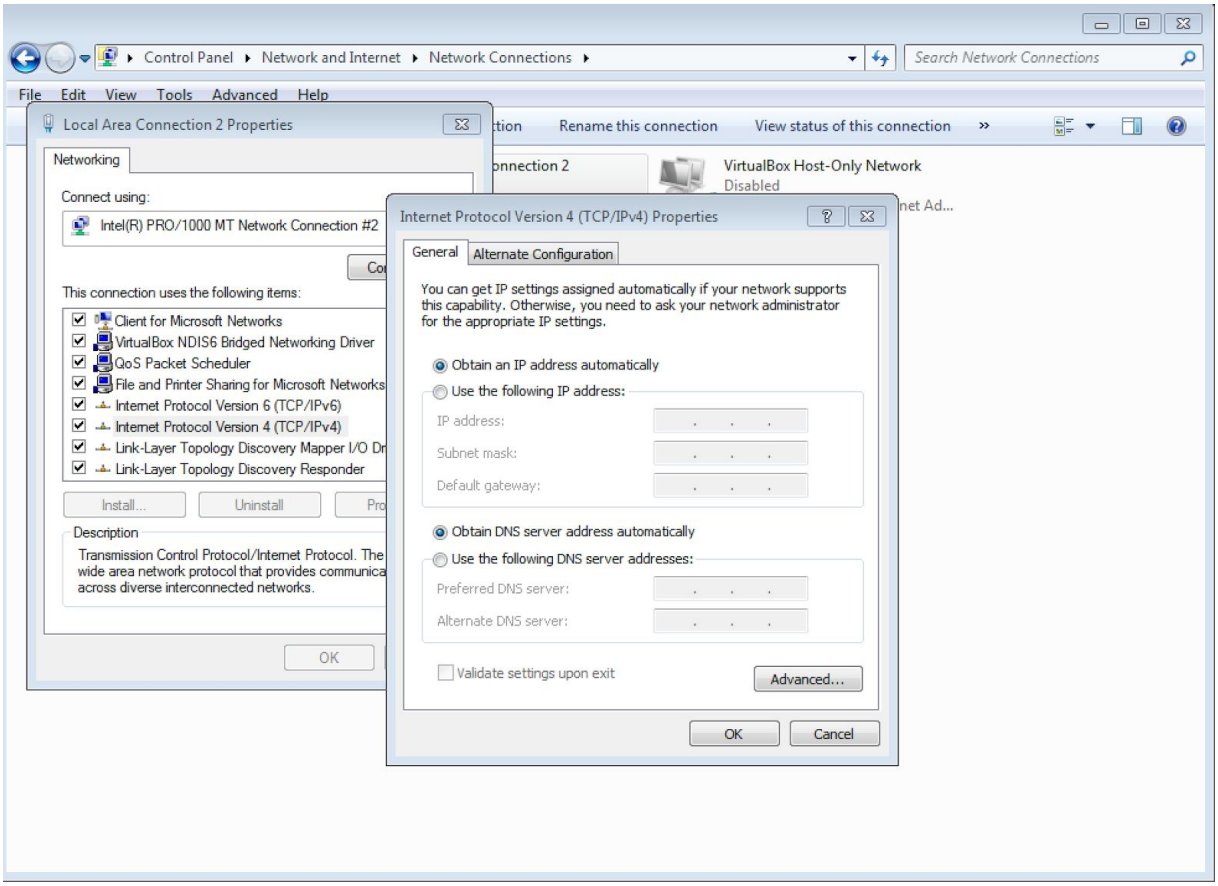


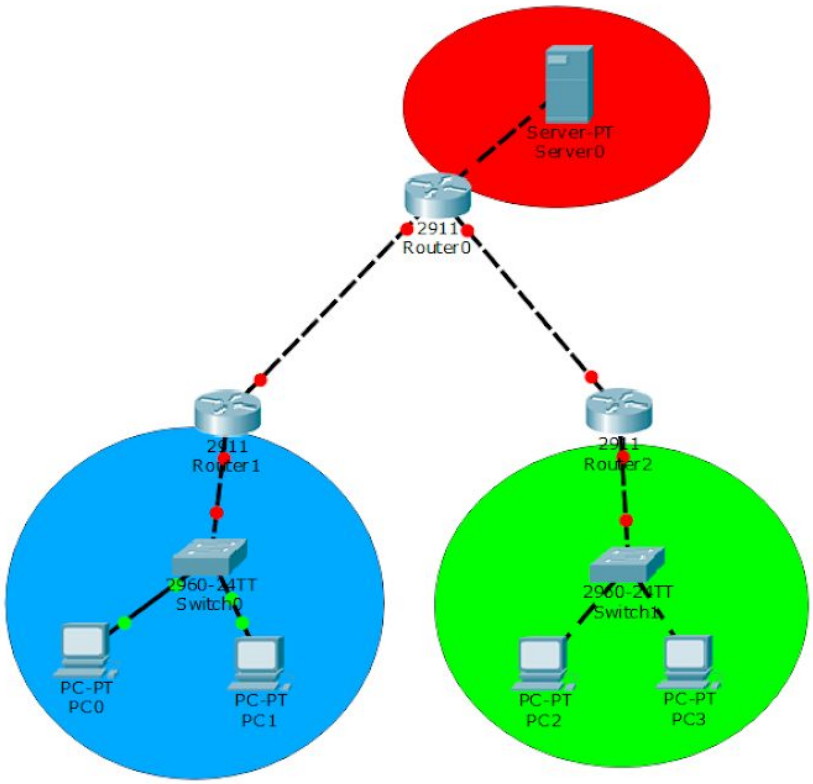
Figura 15.8

Aici vom selecta “**Obtain an IP address automatically**” si “**Obtain DNS server address automatically”**. Aceste 2 optiuni vor spune PC-ului sa trimita o cerere DHCP in retea pentru o adresa IP dinamica.

Laboratorul #11

Acum am ajuns la partea de laborator (partea practica), care o ai inclusa in atasamente. Aici urmeaza sa implementam cele discutate mai sus. Urmareste cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Configurare DHCP in retele LAN si recapitulare protocoale de rutare.



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

2) ACL

Un **ACL (Access Control List)** reprezinta un set de reguli cu scopul de a bloca sau permite accesul dintr-o retea la o anumita resursa. Aceste reguli sunt setate pe Routere sau pe Firewall-uri.

“ACL-urile stau la baza *conceptului de securitate* (limitare a accesului) intr-o sau dintr-o retea (ex: Din Internet in reteaua Interna - LAN sau invers).

Gandeste-te la acest concept, ca la un Bodyguard care sta la intrarea unui club in care se organizeaza o petrecere privata. Acesta va avea o lista cu toti invitatii la acea petrecere. Pe masura ce oamenii incearca sa intre in locatie, bodyguard-ul il va verifica pe fiecare in parte; se va uita pe lista (**ACL**) si va decide pentru fiecare persoana daca are voie in club sau nu. Practic daca te afli pe lista vei fi lasat sa intrii (permit) la petrecere, iar daca nu apari nu vei avea acces (**deny**) inauntru.

Pentru inceput trebuie sa **cream** aceste **reguli** si sa le **includem in ACL**. Dupa cum vom vedea mai jos, aceste **reguli pot varia**: de la *permiterea unei retele intregi* sa acceseze o alta retea, la *permiterea sau respingerea accesului a unui singur PC* la un server pe un anumit port (ex: SSH - 22, Web - 80).

Dupa ce cream o astfel de lista de acces si adaugam reguli de ***permit*** *sau* ***deny,*** trebuie sa o punem in functie. Mai exact, trebuie sa **alegem o interfata** a Router-ului (sau a unui Firewall) pe care dorim sa o setam si **directia** (**IN/OUT**) in care vrem sa facem aceasta filtrare. Exista 2 tipuri principale de ACL-uri:

- **ACL Standard**
- **ACL Extended**

a) ACL Standard

Scopul ACL-urilor de tip Standard este sa faca filtrarea traficului dupa ***IP-ul sursa*** **!**

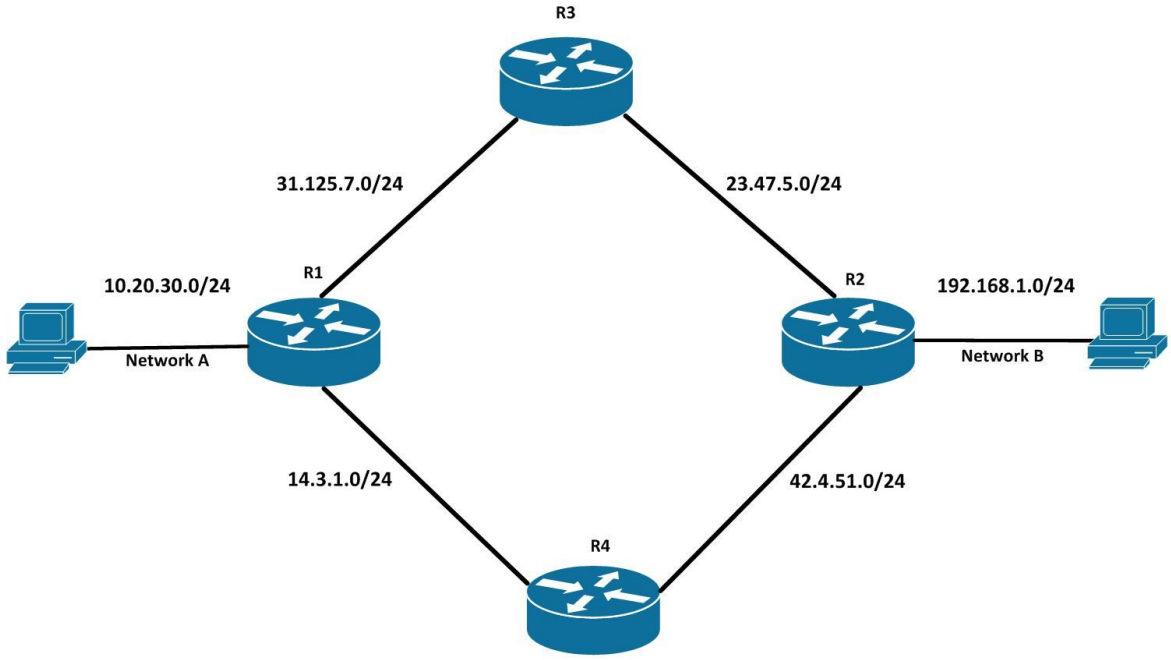


Figura 15.9

Sa spunem ca (din motive de securitate) PC-ului din reteaua A, cu IP-ul 10.20.30.8, nu ii vom da voie sa iasa din reteaua LAN. Astfel tot ce trebuie sa facem este sa cream o lista de acces in care sa specificam acest lucru. Regulile acestei liste vor arata astfel:

```
#deny host 10.20.30.8
```

```
#permit any
```

Aceasta regula va fi setata pe R2, pe **interfata cea mai apropiata** de server (in cazul acesta, cea direct conectata la server) in **directia OUT**.

```
R2(config)#
R2(config)#ip access-list standard ACL_STD
R2(config-std-nacl)#deny host 10.20.30.2
R2(config-std-nacl)#permit any
R2(config-std-nacl)#exit
R2(config)#
R2(config)#interface Gi2/0
R2(config-if)#ip access-group ACL_STD in
R2(config-if)#
```

Figura 15.10

Am adaugat cea de a 2-a linie (#**permit any**) pentru ca, by default, ***la finalul fiecarui ACL apare o regula "implicita de deny"*** (#**deny any**). Noi dorim sa oprim doar traficul de la PC catre orice alta destinatie si sa **permitem in rest orice alt tip de trafic**.

Si iata rezultatul (in figura de mai jos) in urma testarii de pe mai multe dispozitive printre care si PC-ul (poti vedea ca au fost blocate 20 de pachete, care proveneau de la acesta). Verificam folosind comanda:

```
#show ip access-list
```

Figura 15.11

b) ACL Extended

Scopul ACL-urilor de tip Extended este sa faca filtrarea traficului dupa:

- ***IP Sursa***
- ***IP Destinatie***
- ***Port Sursa***
- ***Port Destinatie***
- ***Protocol (IP, TCP, UDP etc.)***

Astfel, acest tip de liste ne ofera o flexibilitate mult mai mare cand vine vorba de control. Putem controla orice flux de trafic indiferent de sursa, destinatie si aplicatie folosita.

Sintaxa pentru ACL-urile **Standard:**

#ip access-list standard *NUME*

#permit *ip_sursa wildcard_mask*

Sintaxa pentru ACL-urile **Extended:**

#ip access-list extended *NUME*

#permit protocol *ip_sursa wildcard_mask port_sursa ip_destinatie wildcard_mask port_destinatie*

Setarea ACL-urilor pe Interfete

Cisco ne recomanda urmatoarele:

- ***ACL-urile standard*** se configureaza cat mai aproape de **destinatie**
- ***ACL-urile extended*** se configureaza cat mai aproape de **sursa**

```
#interface Gig 0/1
```

```
#ip access-group NUME_ACL [in/out]
```

Alte exemple (ACL Standard):

- `#deny 192.168.99.0 0.0.0.255`
  - //va permite tot range-ul /24

- `#permit 85.1.245.5 0.0.0.0`
  - //va permite doar acest IP

- `#deny 172.16.0.0 0.0.127.255`
  - //va permite reteaua 172.16.0.0/17

Alte exemple (ACL Extended):

- `#permit ip 172.30.0.0 0.0.0.255 any`
  - //permite traficul de la sursa 172.30.0/24 catre orice destinatie

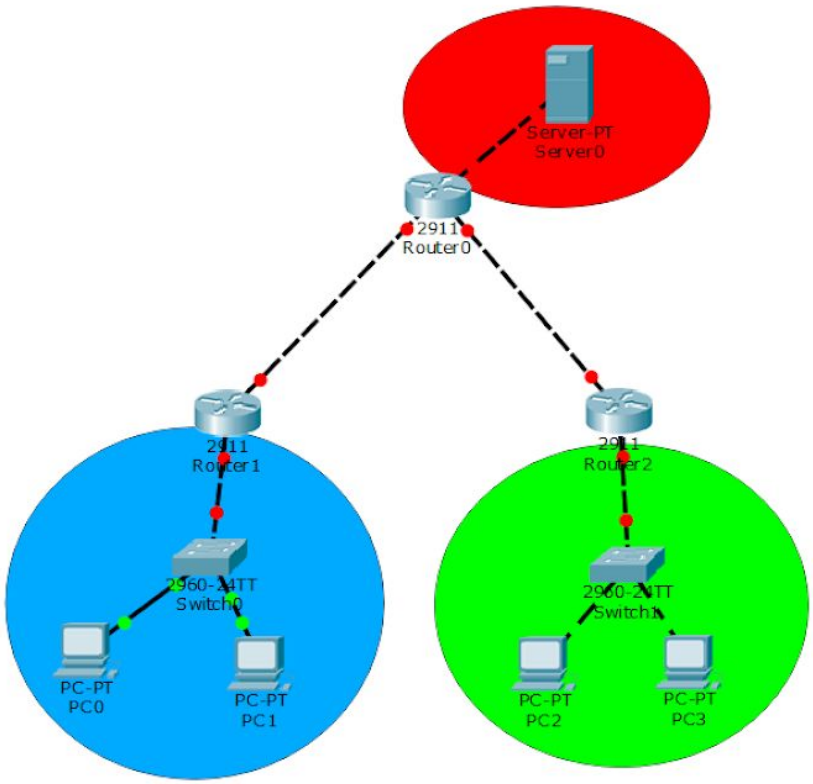
- `#permit tcp host 10.45.21.5 eq 22 any`
  - //permite orice trafic SSH de return de la 10.45.21.5

- `#deny udp 172.16.0.0 0.0.0.255 85.98.2.0 0.0.254.255 eq 53`
  - //blocheaza traficul DNS (port 53 UDP) de la reteaua 172.16.0.0/24 la 85.98.2.0/23

Laboratorul #12

In acest laborator vom configura conectitivatea de baza intre Routere, dupa care putem trece la implementarea ACL-urilor (atat celor Standard cat si Extended). Te invit sa urmaresti cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Filtrarea traficului din retea pe baza unui set de reguli. Configurare ACL-uri.



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

3) NAT

Cum functioneaza NAT ?

Organizatiile care "administreaza" Internetul au propus, prin conventie, adresele IP Private sa nu poata fi rutate in Internet (**orice pachet cu IP-ul sursa Privat va fi aruncat !**).

Astfel toate companiile furnizoare de servicii de Internet au implementat politici de filtrare a traficului (ACL) pe baza IP-ului sursa care verifica daca un pachet are un IP este privat sau nu. In cazul in care, IP-ul sursa este privat (i.e: 10..., 172.16... sau 192.168...) atunci acesta va fi oprit si "aruncat la gunoi", transportarea lui catre destinatie nefiind permisa.

AICI intervine NAT:

*Network Address Translation (**NAT**) mascheaza ("translateaza") un IP Privat intr-un IP Public.*

Practic fara acest mecanism nu am putea accesa internetul. **De NAT se ocupa Routerul** (fie ca este vorba de cel *al companiei tale* sau *a celei la care lucrezi* sau ca este vorba de *Routerul Wireless* din sufrageriile noastre)

Tipurile de NAT

Exista mai multe tipuri de NAT printre care identificam:

1. **NAT Static**
2. **NAT Dinamic**
3. **PAT (Port Address Translation)**

1) NAT Static

Face o **mapare 1-la-1** a unui IP Privat intr-un IP Public.

PC2: **192.168.1.5 -> 42.4.51.8**

Este folosit, de obicei, in momentul in care avem un server (Web, FTP, etc.) in reteaua locala (**LAN**) si dorim ca resursele de pe acel server (pagina Web, Serverul de CS, un fisier, etc.) sa fie accesibile din Internet.

Exemplu: Ai creat un folder cu poze din ultima vacanta pe care doresti sa le impartasesti cu prietenii si familia ta. Te-ai gandit sa apelezi la un server web si pentru ca ai o adresa IP Public in plus de la Furnizorul de Internet, ai decis sa apelezi la NAT Static.

Adresa IP a serverului tau este 192.168.1.5, iar cea publica este 93.1.8.6. Faci setarile pe Routerul tau de acasa (din Interfata Web - Browser) si le trimiti prieteniilor si familiei link-ul http://93.1.8.6/poze_vacanta2016, iar acestia vor putea sa iti vada cu succes pozele :)

2) NAT Dinamic

Face o **mapare m-la-n** a unui IP Privat intr-un IP Public, unde m nu este neaparat egal cu n.

PC1: **192.168.1.6 -> 23.47.5.7**

PC2: **192.168.1.7 -> 23.47.5.8**

PC3: **192.168.1.8 -> 23.47.5.9**

NAT-ul dinamic *foloseste un spatiu de adrese* (ex: 93.1.8.7 pana la 93.1.8.10) pe care le poate aloca cate unui singur calculator care doreste sa ajunga in Internet. Functioneaza pe principiul **FIFO** (primul venit, primul servit), asadar daca avem **20** de PC-uri in retea si numai **4 adrese IP** publice disponibile, **doar 4 din cele 20 de PC**-uri vor putea ajunge (comunica) in Internet.

3) PAT (Port Address Translation)

In cazul PAT, **maparea este n-la-1**. Adica, avem mai multe adrese IP Private si te "transformam" intr-o singura adresa IP Publica la care **adaugam Portul Sursa** al conexiunii.

PC1: **192.168.1.6:22413 -> 23.47.5.5:22413**

PC2: **192.168.1.7:62459 -> 23.47.5.5:62459**

PAT ascunde mai multe device-uri (cu IP Privat) in spatele unui singur IP Public.

O conexiune dintre 2 device-uri in Internet contine si urmatoarele elemente:

- *IP Sursa*
- *IP Destinatie*
- *Port Sursa*
- *Port Destinatie*

Cand vine vorba de PAT, Routerul va folosi adresa **IP Sursa** (Privata) si **Portul Sursa**, pentru a identifica conexiunea (exact cum este ilustrat in exemplul de mai sus).

PAT este cea mai folosita varianta de NAT, fiind configurat pe marea majoritate a Routerelor Home-Oriented (TP-Link, D-Link, Asus, Huawei, Cisco etc.).

Configurare NAT pe Routere

Acum ca stim ce este NAT, cum functioneaza acesta si cate tipuri sunt, propun sa trecem la partea de configurare pe Routere:

a) NAT Static

NAT-ul Static reprezinta o simpla mapare intre o adresa IP privata si una publica:

```
R2(config)#ip nat inside source static 192.168.1.2 42.4.51.9
```

La sfarsit trebuie sa pornim NAT-ul pe interfete:

```
R2(config)#interface Gi0/1
R2(config-if)#ip nat inside

R2(config)#interface Gi0/2
R2(config-if)#ip nat outside
```

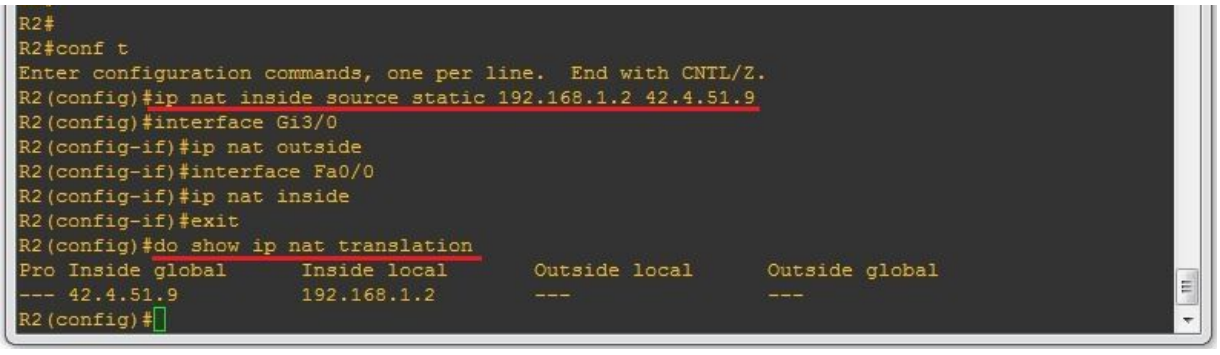


Figura 15.12

Dupa cum poti vedea in ambele figuri comanda de verificare ne afiseaza “maparea” pe care noi am facut-o. In figura de mai jos am dat ping (de pe host-ul nostru pe Ubuntu) catre un PC care e legat la R2. Se poate vedea IP-ul sursa (10.20.30.2 - Ubuntu) si IP-ul destinatie public (42.4.51.9 - cel al PC-ului) careia i se face translatarea in IP-ul privat 192.168.1.2.

```
R2#show ip nat translation
```

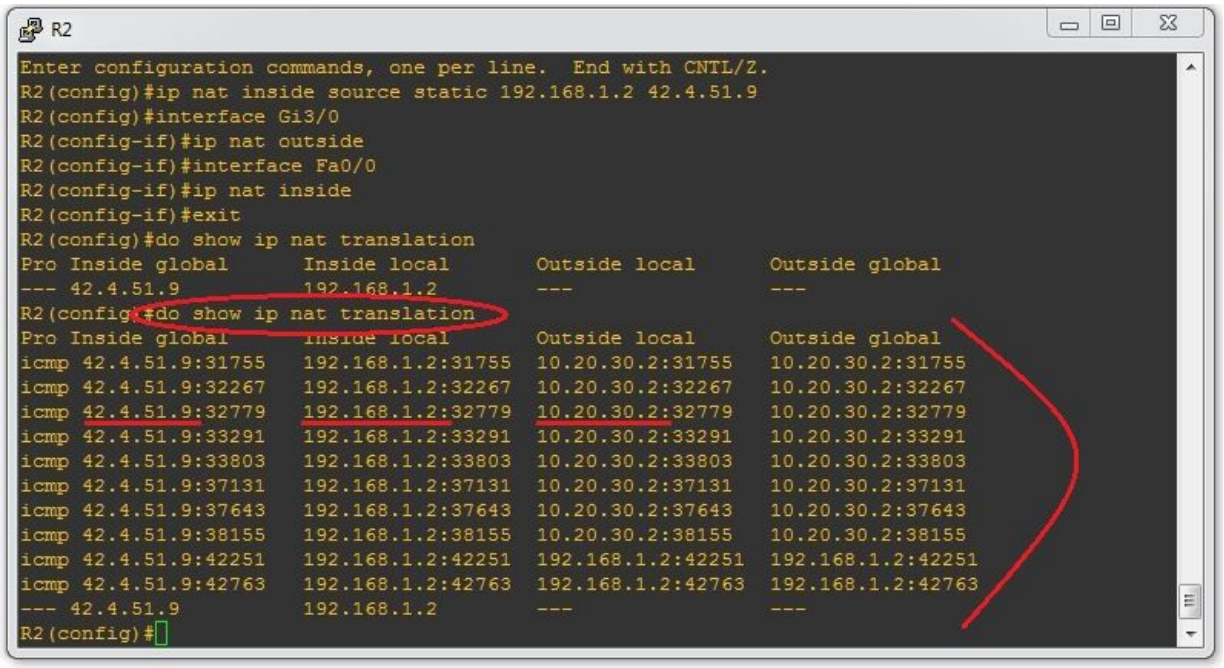


Figura 15.13

b) NAT Dinamic

Pentru a configura NAT-ul Dinamic avem nevoie sa cream o lista (ACL) care sa identifice adresele IP care se doresc a fi “NATate”:

```
R1(config)#ip access-list standard NAT_ACL
R1(config-acl)#permit 10.20.30.0 0.0.0.255
```

Dupa care trebuie sa configuram o “piscina” (pool) de adrese (publice) pe care dorim sa le alocam:

```
R1(config)#ip nat pool 31.125.7.11 31.125.7.14 netmask 255.255.255.0
```

La sfarsit vom creea regula prin care specificam ACL-ul si pool-ul creat mai devreme:

```
R1(config)#ip nat inside source list NAT_ACL pool NAT_IPs
```

Si pornim regulile pe interfete:

```
R1(config)#interface Gi0/1
R1(config-if)#ip nat inside

R1(config)#interface Gi0/2
R1(config-if)#ip nat outside
```

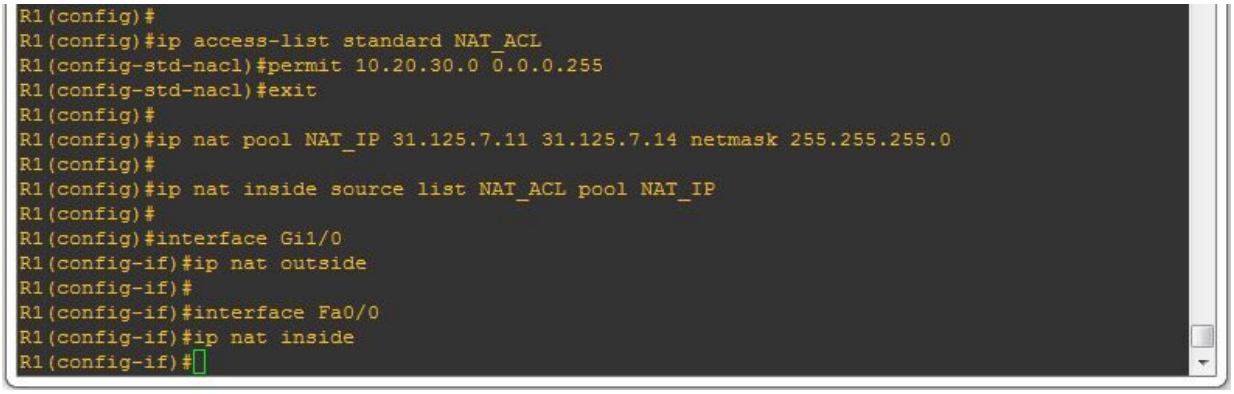


Figura 15.14

In urma comenzii de show, vom vedea (in acest scenariu) un PC, folosind o singura adresa IP:

```
R1#show ip nat translation
```

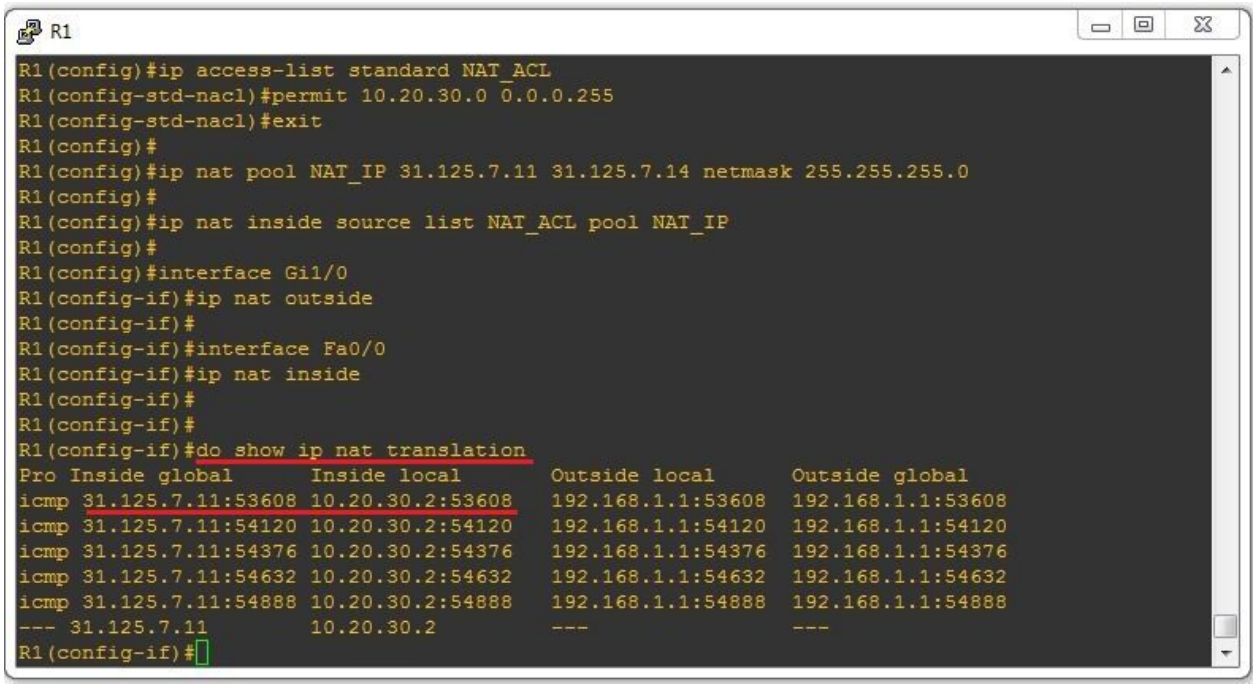


Figura 15.15

c) PAT (**P**ort **A**ddress **T**ranslation**)**

Pentru a configura NAT-ul Dinamic avem nevoie sa cream o lista (ACL) care sa identifice adresele IP care se doresc a fi “NATate”:

```
R1(config)#ip access-list standard NAT_ACL
R1(config-acl)#permit 192.168.1.0 0.0.0.255
```

La sfarsit vom creea regula prin care specificam ACL-ul si interfata pe care dorim sa facem PAT-ul, urmat de keyword-ul *overload* (care practic spune foloseste aceeasi adresa IP in mod repetat):

```
R1(config)#ip nat inside source list NAT_ACL interface Gi0/2 overload
```

Si pornim regulile pe interfete:

```
R1(config)#interface Gi0/1
R1(config-if)#ip nat inside

R1(config)#interface Gi0/2
R1(config-if)#ip nat outside
```

```
R2(config)#
R2(config)#ip access-list standard NAT_ACL
R2(config-std-nacl)#permit 192.168.1.0 0.0.0.255
R2(config-std-nacl)#exit
R2(config)#
R2(config)#ip nat inside source list NAT_ACL interface Gi2/0 overload
R2(config)#
R2(config)#interface Gi2/0
R2(config-if)#ip nat outside
R2(config-if)#
R2(config-if)#interface Fa0/0
R2(config-if)#ip nat inside
R2(config-if)#
```

Figura 15.16

`R1#show ip nat translation`

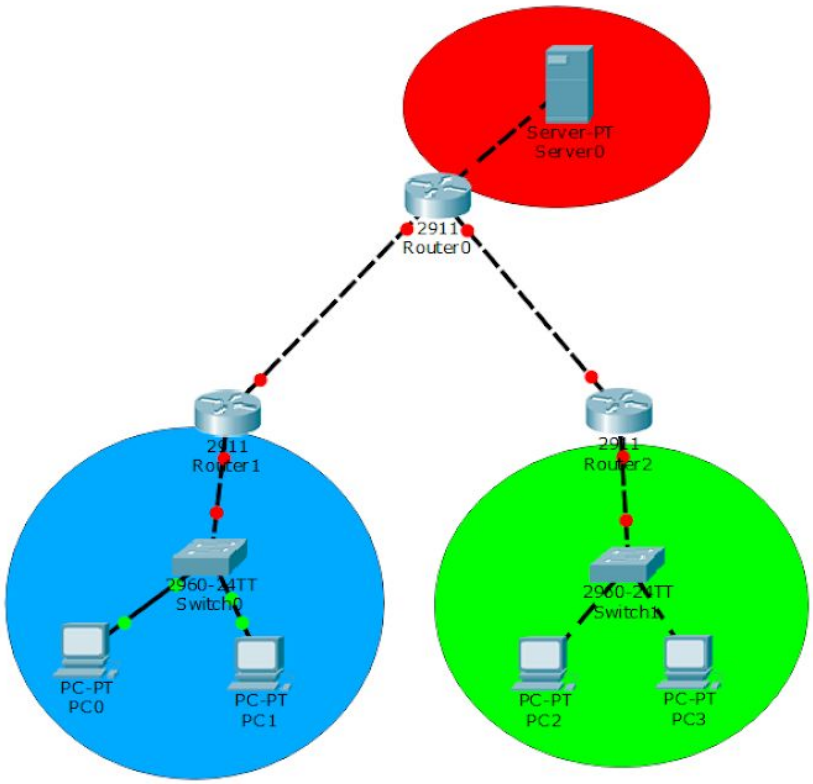
```
R2#
R2#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 23.47.5.1:1027    192.168.1.1:1      23.47.5.2:1        23.47.5.2:1027
icmp 23.47.5.1:1026    192.168.1.2:0      23.47.5.2:0        23.47.5.2:1026
udp 23.47.5.1:4502     192.168.1.2:11280  10.20.30.1:11281   10.20.30.1:11281
udp 23.47.5.1:4501     192.168.1.2:34352  10.20.30.1:34353   10.20.30.1:34353
R2#
```

Figura 15.17

Laboratorul #13

Laboratorul este similar cu cel de data trecuta (de la ACL-uri) dar de data aceasta vom configura NAT pe Routerele existenta in retea. Te invit sa urmaresti cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aplicarea conceptelor de NAT intr-un mediu care simuleaza Internetul.



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Descarca folderul cu laboratoare de AICI.

Capitolul 16 - IPv6

Ce este IPv6 ?

IPv6 reprezinta noul mod (forma de adresare) de comunicare in Internet, odata cu terminarea adreselor **IPv4** (in numar de 4,3 miliarde). Adresele IPv6 sunt intr-un numar seminificativ mai mare, acestea avand un total de 128 de biti pentru exprimarea adreselor (spre deosebire de cei 32 de biti al adreselor IPv4). Dupa cum spuneam, spatiul de adresare este pe 128 biti permite un numar mult, mult, mult mai mare de adrese si subretele.

"In total, pe IPv6 sunt disponibile 340.282.366.920.938.463.463.374.607.431.768.211.456 adrese, echivalentul a 340 trilioane de trilioane de trilioane de adrese".

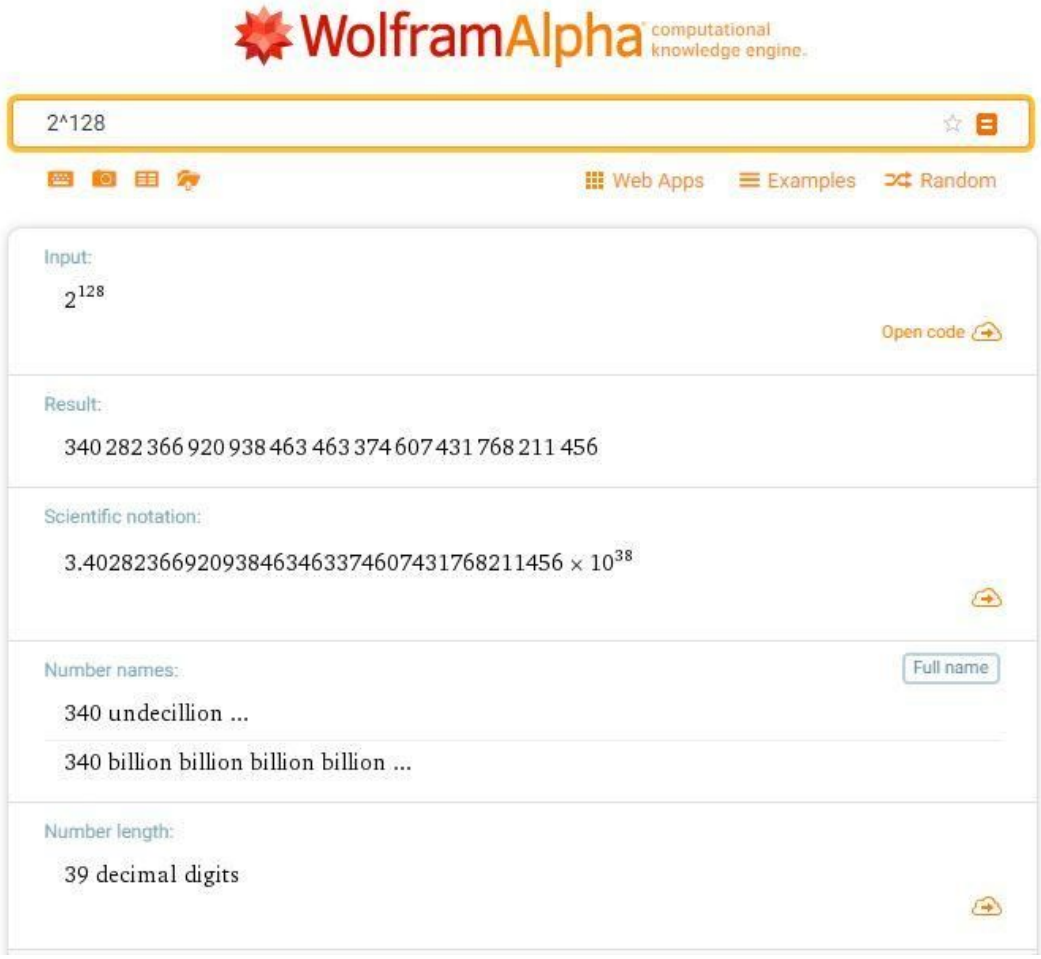


Figura 16.1

O adresa IPv6 este exprimata in Hexa (16 valori, intre 0 - 9 si A - F) si poate arata in felul urmator:

2003:4581:A7C1:EFDB:0000:0000:1327:0001

Dupa cum poti vedea difera cu mult fata de o adresa IPv4 (ex: 10.87.12.1) cu care suntem obisnuiti.

Simplificarea adreselor IPv6

Cei care au dezvoltat IPv6 si-au dat seama ca adresele pot fi foarte lungi, greu de retinut/scris si s-au gandit la o solutie, si anume sa le simplifice intr-un anumit mod:

1) In cazul in care avem o secvente de 2 sau mai multe campuri de 0 (0000:0000...) acestea poti fi simplificate cu **::**

1234:ABCD:3123:0000:0000:0000:0000:0000 -> 1234:ABCD:3123::

De exemplu, **ruta default pe IPv6** este exprimata astfel **::/0** fata de 0.0.0.0/0 cum este cea de pe IPv4.

2) In cazul in care nu avem posibilitatea de a simplifica atat de drastic, mai avem o solutie: **scriem un grup de 0000 ca fiind 0**. Iata un exemplu:

1234:ABCD:3123:00A8:0A31:8000**:0000**:0001-> 1234:ABCD:3123:00A8:0A31:8000**:0:**001

3) Exemplul de mai sus il putem simplifica si mai multe, eliminand 0-urile din fata unui camp:
1234:ABCD:3123:00A8:0A31:8000**:0000**:0001 -> 1234:ABCD:3123:A8:A31:8000**:0:1**

In exemplul de mai sus, poti vedea cum **am eliminat toate 0-urile de la inceput:**

00A8 -> A8, 0001 -> 1

Adresa IPv6 din exemplul de la punctul anterior (*2003:4581:A7C1:EFDB:0000:0000:1327:0001)* se mai poate scrie și sub următoarea forma: 2003:4581:A7C1:EFDB::1327:1

Tipuri de adrese IPv6

Acum ca ai inteles ce este IPv6 si cum arata o astfel de adresa, este timpul sa vorbim despre tipurile de adrese IPv6:

I. **Globale** - 2000::/3 (*folosite in Internet*)
II. **Link Local** - FE80::/10 (*folosite in LAN*)

Haide sa incepem cu primul tip de adrese IPv6 si anume cel Globale:

1) Adrese IPv6 Globale

Adresele **IPv6 Globale** sunt similare cu adresele **IPv4 publice** (**pot** fi **folosite** in **Internet** si nu doar in LAN). Aceste adrese globale pot sa inceapa cu cifra **2** sau cu **3** (ex: 2000::/3 este range-ul desemnat pentru ele) si pot avea orice valoarea in rest. Iata in figura de mai jos o structura a acestui tip de adrese IPv6:

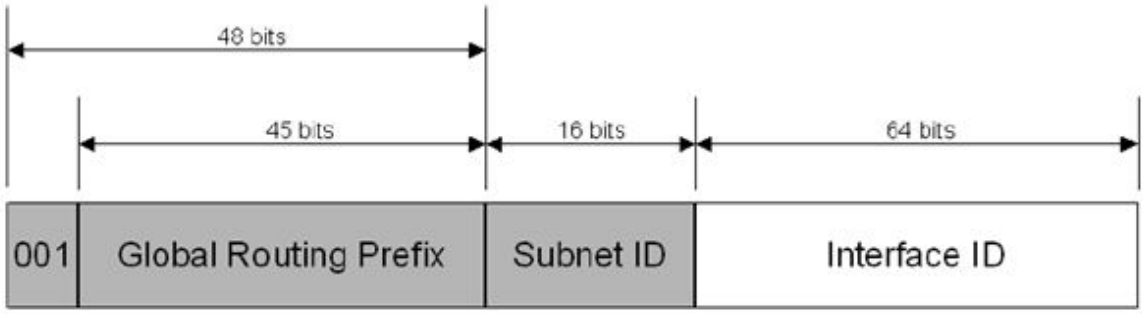


Figura 16.2

In primul rand, **primii 3 biti sunt rezervati** (astfel adresele globale incep cu **2**), dupa care urmeaza Global Routing Prefix (sau spatiul de adrese care i-a fost alocat unui ISP). Subnet ID-ul reprezinta ID-urile retelelor (care pot fi in numar de 65536 = 2^16), iar la sfarsit avem Interface ID care se refera la acea adresa unica rezervata unui dispozitiv din retea. Iata cateva exemple de adrese IPv6 globala (ATENTIE: acestea sunt adrese de host si nu adrese de retea):

- **2001:DB8:A0B:12F0::1**/64
- **3731:ADE0:9923:23::90**/64
- **2020:ABCD:1:FFF0:84:ADEF**/64

2) Adrese IPv6 Link Local

Sunt adrese speciale cu scopul de a face posibila comunicare intre dispozitivele (PC-uri, Laptop-uri, Smartphone-uri etc.) din reteaua locala (LAN). Avantajul acestor tipuri de adrese este faptul ca se configureaza automat (**autoconfig**) si nu necesita o interventie din partea utilizatorilor. Adresele IPv6 de tipul Link Local sunt in formatul FE80::/10. Adica incep cu FE80 iar restul adresei se **genereza automat** prin tehnica **EUI-64**. Cel mai probabil daca (folosesti Windows) te uiti acum in CMD, vei putea vedea o adresa IPv6 Link Local:

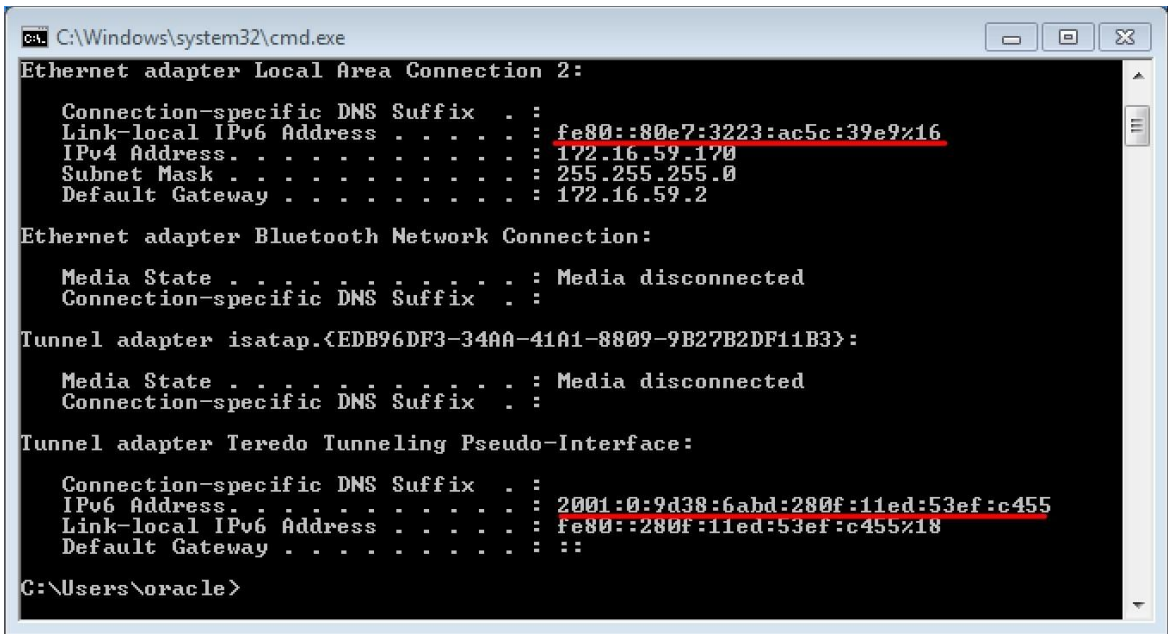


Figura 16.3

Prima adresa IPv6 (subliniata cu rosu) este adresa Link-Local pentru interfata Ethernet (cea prin cablu), iar cea de a 2-a adresa IPv6 care incepe cu 2001:0... este o adresa **globala** care apartine interfetei Tunnel. Dupa cum spuneam si mai devreme, ele sunt generate automat folosind EUI-64. Aceasta tehnica este folosita de majoritatea vendorilor si OS-urilor (Cisco, Juniper, Ubuntu etc.), dar fara Windows. Microsoft a ales ca incepand cu versiunea Vista sa genereze aleatoriu (random) aceste adrese Link-Local. Pentru majoritatea sistemelor: **EUI-64** este o tehnica care foloseste adresa **MAC** (pentru ca este considerata **unica** pentru fiecare dispozitiv in parte) si un camp FFFE pentru a face asta. In Figura 2.4 de mai jos poti vedea exact cum se procedeaza:

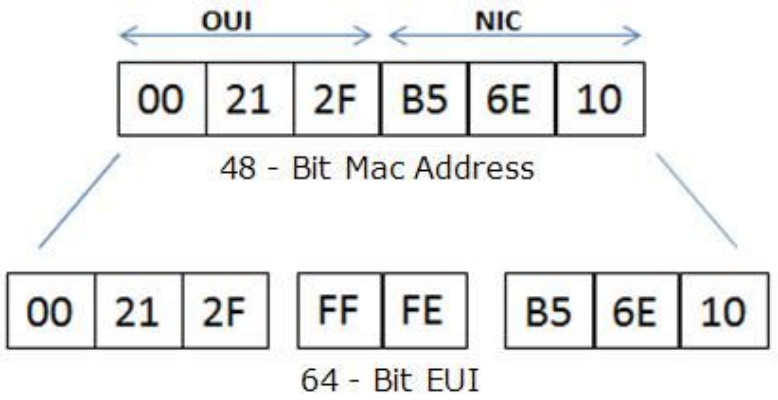


Figura 16.4

Adresa **MAC initiala** *00:21:2F:B5:6E:10* este **impartita** in doua, iar la mijloc se adauga **FF:FE.** Astfel, *0021:2FFF:FEB5:6E10* vor fi bitii de host pentru adresa IPv6:

FE80::0021:2FFF:FEB5:6E10/64

In unele implementari se modifica (neaga) al 2-lea bit. Adica daca al 2-lea bit are valoarea 1 el va deveni 0, daca are valoarea 0 va deveni 1.

In acest caz adresa IPv6 ar arata in felul urmator:

FE80::**0021**:2FFF:FEB5:6E10/64 -> FE80::**0023**:2FFF:FEB5:6E10/64

Inainte sa trecem mai departe vreau sa fac o scurta precizarea si anume: **adresele IPv6 Link-Local NU au voie sa treaca (sa iasa) din reteaua locala**.

3 Moduri de transmitere a pachetelor

Spre deosebire de IPv4, in IPv6 nu exista mesaje de tipul BROADCAST. Mesajele Broadcast au fost inlocuite cu cele de tipul Multicast. Daca stam putin sa ne gandim, e logic pentru ca daca vreau sa trimit unui grup specific de device-uri din retea (aka. Multicast), atunci pot folosi acelasi pachet **sa transmit tuturor device-urilor** din retea (astfel grupul specific marindu-se incluzandu-le pe toate).

Acum sa vedem modurile prin care pot fi transmise mesajele in IPv6:

- **Unicast** - aka. *one-to-one*
- **Multicast** - aka. *one-to-many* (sau one-to-all)
- **Anycast** - aka. *one-to-closest*

Exemplu Anycast:

Daca avem mai multe servere DNS in Romania, se va trimite un Anycast si va fi folosit cel mai apropiat server de locatia noastra (astfel timpul de raspuns va fi mai scurt si conexiunea mai rapida).

Subnetarea pe IPv6

Ooo, da ! Subnetare. Foarte multa lume este speriata de subnetarea pe IPv6, dar eu le spun mereu ca nu au de ce sa-si faca griji pentru ca subnetarea pe IPv6 este (in opinia mea) mai simpla fata de cea pe IPv4.

Cand vorbim de retelel IPv6, niciodata nu vei vedea: “vreau o retea IPv6 cu 5 adrese sau 10 adrese”... nu. De ce ? Pentru ca sunt suficient de multe adrese disponibile, iar dimensiunea unei retele nu trebuie sa ne ingrijoreze.
De cele mai multe ori vei vedea retele IPv6 care au o masca de /64, /80, /96 sau altele. Cel mai des, pot spune ca am vazut retele /64.

Iar acum sa ne gandim ce inseamna (sau cum arata) o retea /64 ? Sa luam urmatorul exemplu:

2002:ABCD:1234:9FD8::/64

Aceasta adresa reprezinta un inceput foarte bun de la care putem subneta. /64, in acest caz, se refera pur si simplu la faptul ca primii **64 de bit**i **nu se schimba** (adica **primele 4 campuri**). De aici, ne putem aloca retele cu /80 pentru orice retea avem/dorim noi. Sa presupunem ca avem nevoie de 4 retele, fiecare cu un numar diferit de IP-uri, dar toate incadrandu-se in /80.

Prima retea: **2002:ABCD:1234:9FD8:0000::/80** (sau 2002:ABCD:1234:9FD8::/80)
A 2-a retea: **2002:ABCD:1234:9FD8:0001::/80** (sau 2002:ABCD:1234:9FD8:1::/80)
A 3-a retea: **2002:ABCD:1234:9FD8:0002::/80** (sau 2002:ABCD:1234:9FD8:2::/80)
A 4-a retea: **2002:ABCD:1234:9FD8:0003::/80** (sau 2002:ABCD:1234:9FD8:3::/80)

Practic ce am facut mai sus ? Am fost atent **in ce camp ma aflu** (in al 5-lea din cele 8) si **am adaugat 1** de la o retea la alta).

Si ATAT ! Nu are sens sa ne stresam cu cate adrese IP avem nevoie in fiecare retea, ci pur si simplu dintr-un spatiu de adrese asignat de catre ISP (Internet Service Provider) (care poate fi de /64, /80, /96 s.a.m.d.) ne gandim, “de cate retele avem nevoie ?”, si incepem sa impartim acea retea in retele mai mici (de obicei cu o masca mai mare cu un camp. Adica daca primim /64, atunci vom folosi /80).

IPv4 si IPv6. Cum comunicam in Internet cu cele 2 protocoale ?

In marea majoritate a cazurilor vom avea retele in care avem configurat atat IPv4 cat si IPv6. By design, cele 2 protocoale sunt INCOMPATIBILE. Adica un PC cu o adresa IPv4 nu poate comunica cu un PC cu o adresa IPv6. Iar acum vine intrebarea ce facem in aceasta situatie ? Si, bineinteles, avem mai multe solutii la aceasta problema:

- **Dual-Stack**
- **Tunele 6to4** sau **4to6**
- **NAT-PT** (aka **NAT64**)

1) Dual-Stack

Conceptul de Dual-Stack este unul foarte simplu. Pur si simplu configuram 2 adrese IP (una fiind v4, iar cealalta fiind v6) pe echipament (Router, PC, etc.). Astfel, PC-urile pot comunica intre ele (sau cu Internetul) folosind oricare dintre cele 2 protocoale.

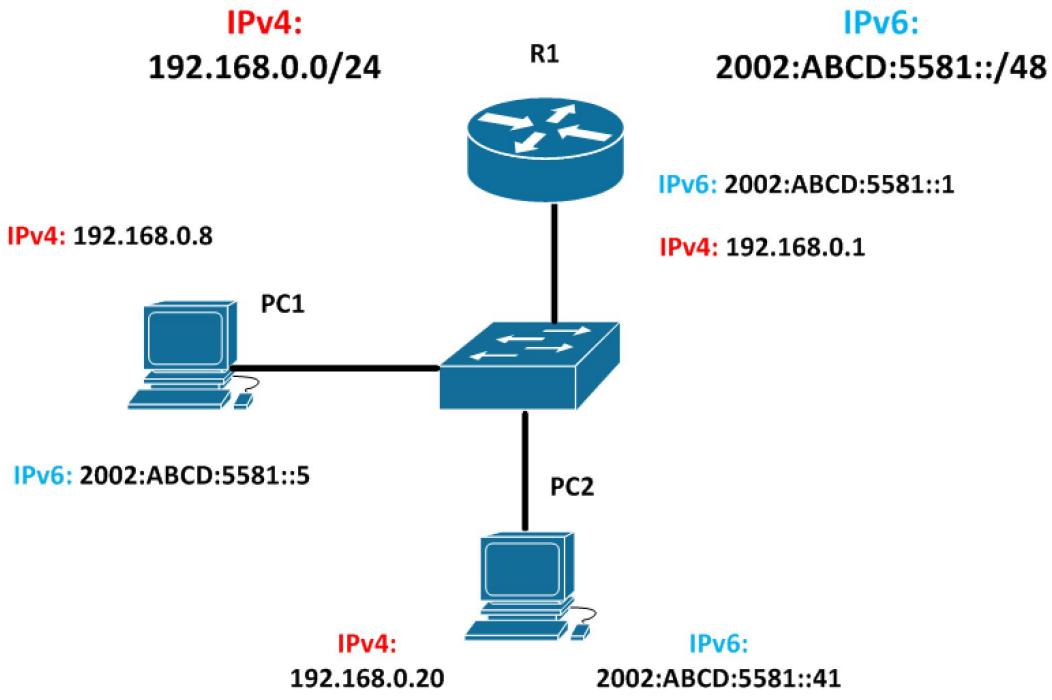


Figura 16.5

2) Tunele 6to4 sau 4to6

Sa presupunem ca avem un scenariu similar cu cel din figura 2.6. 2 retele IPv6 conectate la 2 Routere diferite, in zone geografice diferite, printr-o retea IPv4 (aka Internetul). PC-urile din cele 2 retele nu au cum sa comunice pentru ca o retea de tipul IPv4 nu poate transporta trafic IPv6. O solutie a acestei probleme este implementarea unui **tunel de tipul 6to4**, care va transporta (encapsula) traficul IPv6 intr-un tunel IPv4 peste Internet.

Astfel PC-urile vor putea comunica, de transportul datelor ocupandu-se Routerele. Putem avea chiar si un scenariu opus in care cele 2 Routere au retelele LAN de tipul IPv4, iar reteaua dintre ele este de tipul IPv6. In acest caz avem nevoie de un tunel **4to6**.

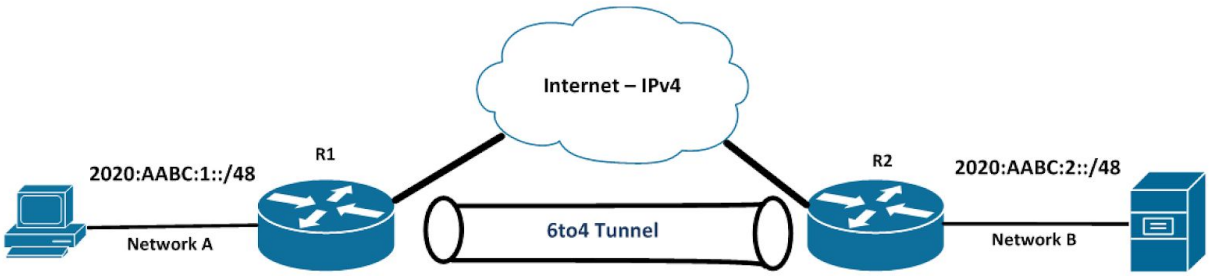


Figura 16.6

3) NAT-PT (**N**etwork **A**ddress **T**ranslation - **Protocol Translation)**

Despre conceptul de NAT vom vorbi in Sectiunea 5 a acestei carti, dar pana atunci iti voi spune pe scurt ce face NAT pentru a intelege si cum functioneaza NAT-PT. NAT este folosit (pe Routere) pentru a translata o adresa Privata (ex: 192.168.1.10) intr-o adresa IP Publica (85.13.217.9) pentru a putea comunica in Internet. Ei bine, NAT-PT face un lucru similar singura diferenta fiind faptul ca acesta, translateaza o adresa IPv4 intr-o adresa IPv6 sau invers (IPv6 -> IPv4).

Spre exemplu un Router poate fi conecta la 2 retele, una folosind IPv4, iar cealalta folosind IPv6. Cele 2 PC-uri nu pot comunica unul cu celalalt (PC-ul din reteaua IPv4 nu poate comunica cu PC-ul din reteaua IPv6). Astfel, Routerul va translata adresa IPv4 intr-o adresa IPv6 si va permite comunicarea intre cele 2, fara probleme.

Iata scenariul de care vorbeam mai devreme in figura 2.7, de mai jos:

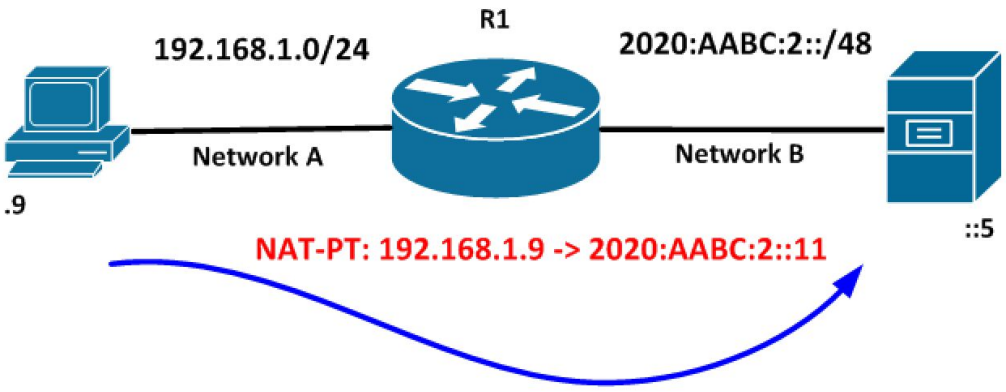


Figura 16.7

Cum Configuram IPv6 ?

Sa trecem la configuram adresele IPv6 pe PC-urile din topologia de mai jos:

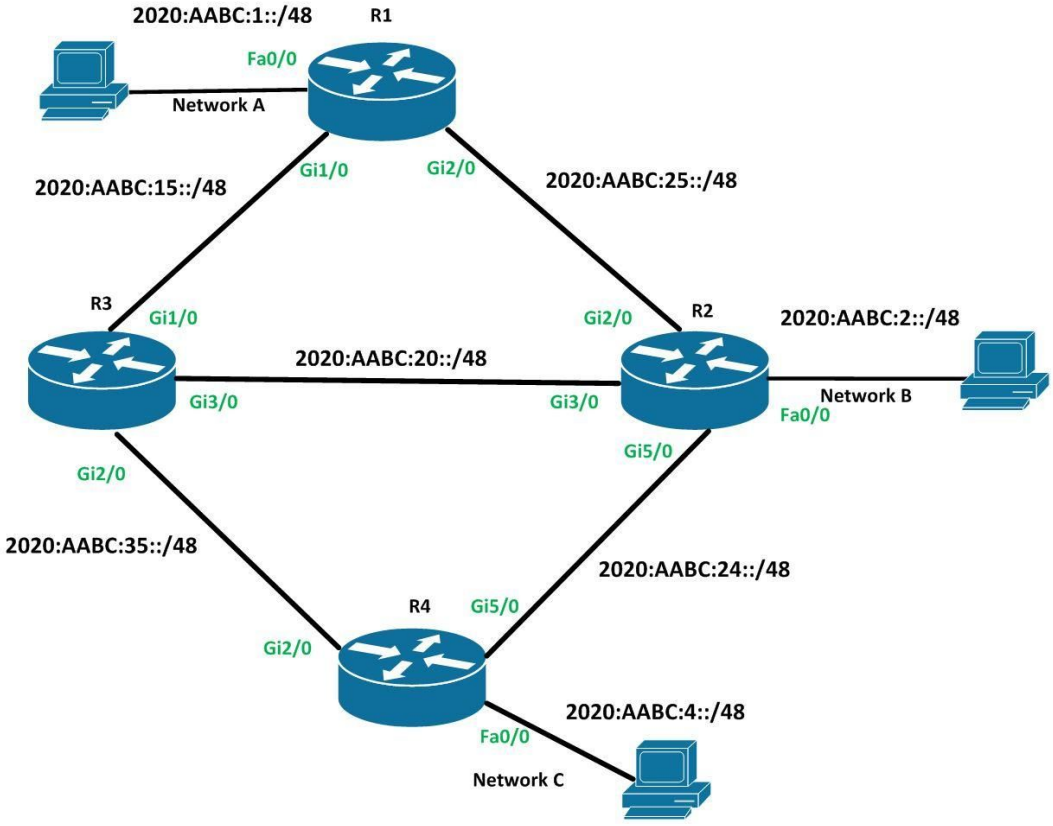


Figura 16.8

Acest procedeu este unul extrem de similar cu cel pentru adresele IPv4:

Mai intai, **vom incepe prin a seta o adresa IPv6 pe Windows 7** (procesul de setare e valabila si pentru Windows 8/10). Procedeul este similar cu cel din primul capitol (in care am setat o adresa IPv4). De obicei adresa IPv6 pe un PC nu trebuie setata (ea este configurata automat), dar putem avea anumite scenarii in care dorim sa setam o adresa IPv6 statica cu scopul de a folosi acel PC ca pe o resursa (ex: server).

Iata cum putem adauga o adresa IPv6 in Windows:

In primul rand trebuie sa ajungem la fereastra in care se afla interfetele noastre de retea.

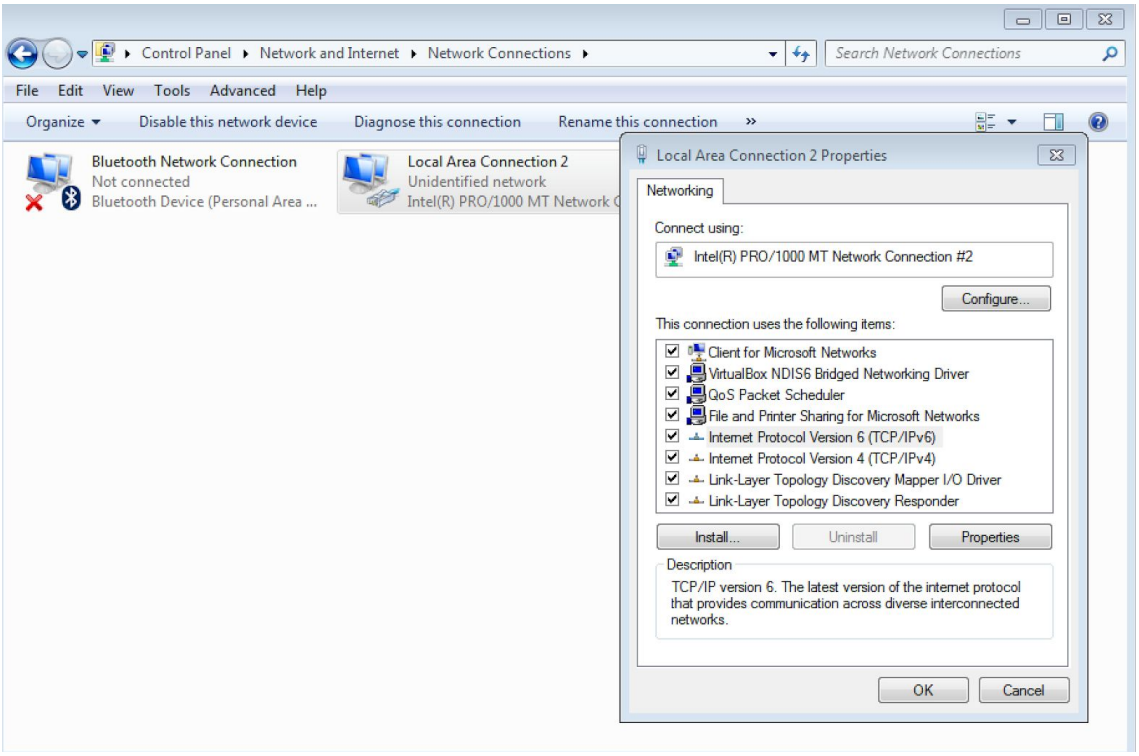


Figura 16.9

Iar aici vom selecta IPv6 si vom apasa pe “**Properties**”, dupa care ne va aparea fereastra in care putem trece adresa IPv6, masca, default gateway-ul iar la sfarsit adresa serverului DNS.

Iata si setarile in figura de mai jos:

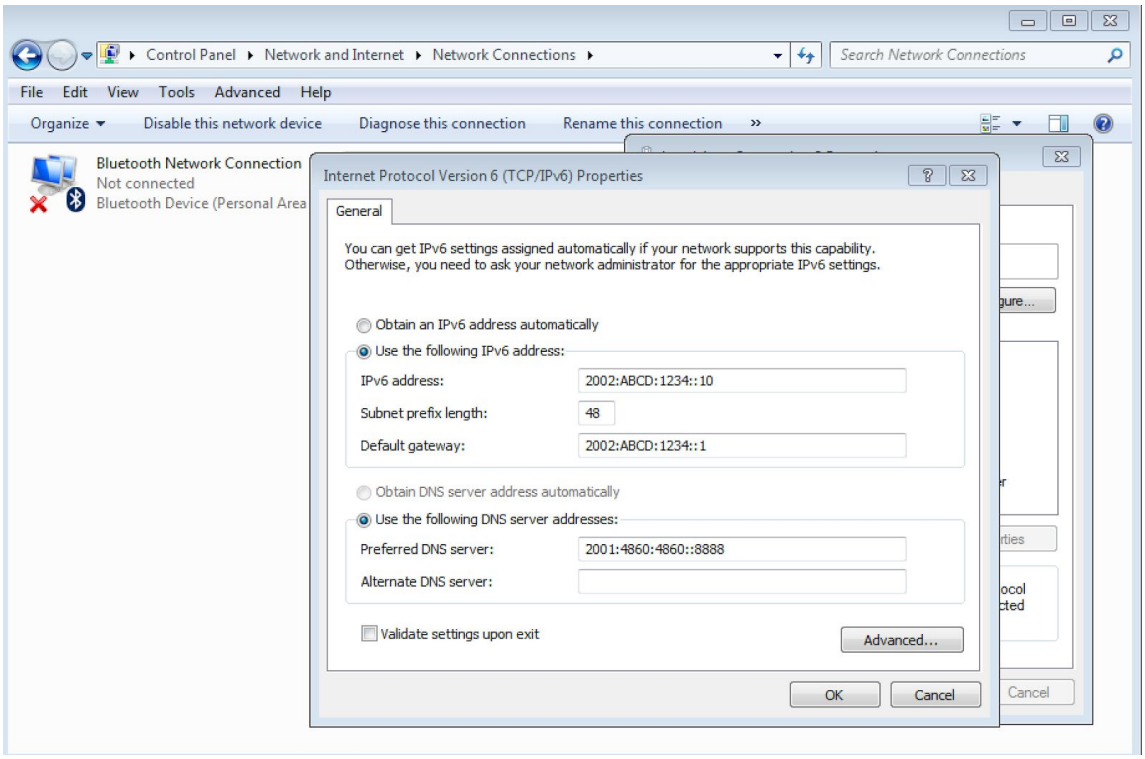


Figura 16.10

Iar acum sa verificam ce am facut mai devreme, folosind comanda **>ipconfig**

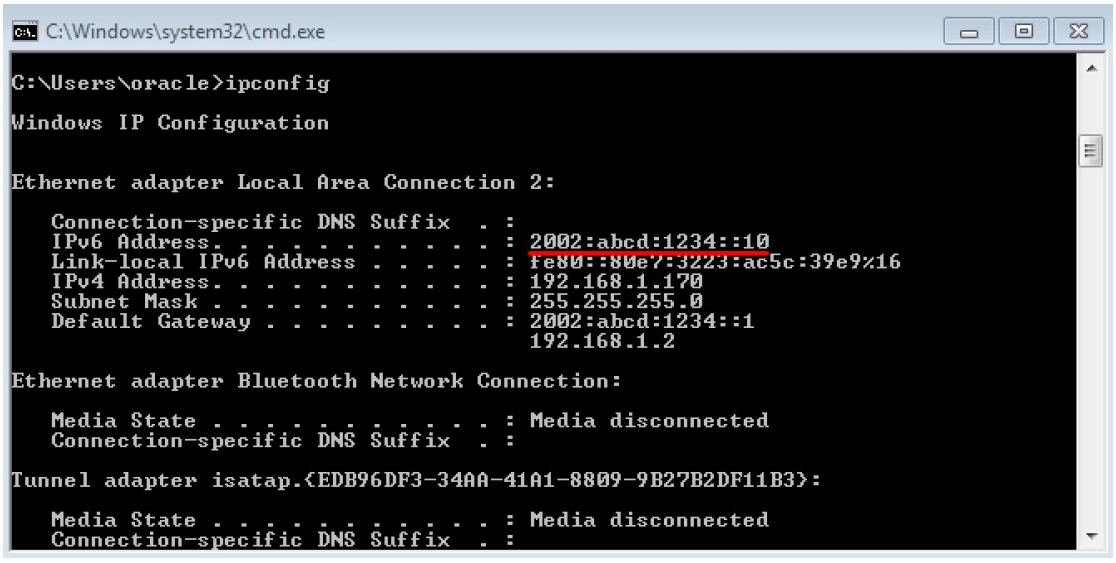


Figura 16.11

Acum, dupa ce vazut cum putem seta o adresa IPv6 pe Windows, a venit momentul sa trecem mai departe cu configul pe Routere:

Pe R1:

```
R1(config)#ipv6 unicast-routing

R1(config)#interface Fa0/0
R1(config-if)#ipv6 address 2020:AABC:1::1/48
R1(config-if)#no shutdown

R1(config)#interface Gi1/0
R1(config-if)#ipv6 address 2020:AABC:15::1/48
R1(config-if)#no shutdown

R1(config)#interface Gi2/0
R1(config-if)#ipv6 address 2020:AABC:25::1/48
R1(config-if)#no shutdown
```

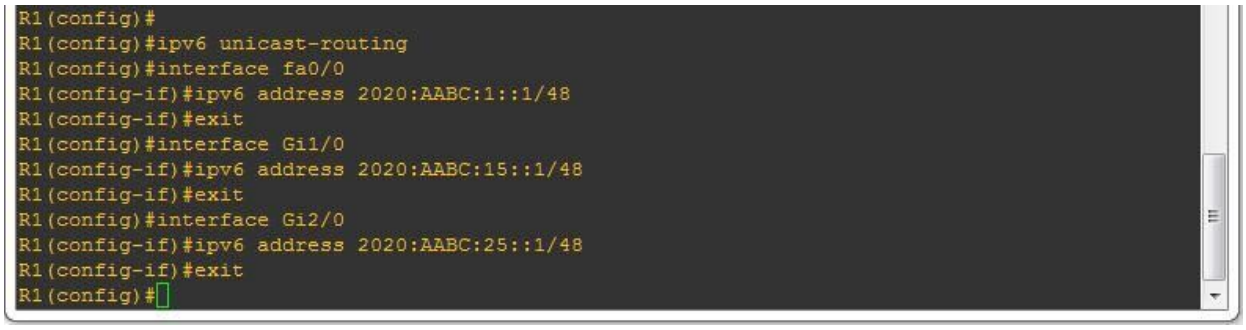


Figura 16.12

Pe R2:

```
R2(config)#ipv6 unicast-routing

R2(config)#interface Gi1/0
R2(config-if)#ipv6 address 2020:AABC:15::2/48
R2(config-if)#no shutdown

R2(config)#interface Gi2/0
R2(config-if)#ipv6 address 2020:AABC:35::1/48
R2(config-if)#no shutdown

R2(config)#interface Gi3/0
R2(config-if)#ipv6 address 2020:AABC:20::1/48
R2(config-if)#no shutdown
```

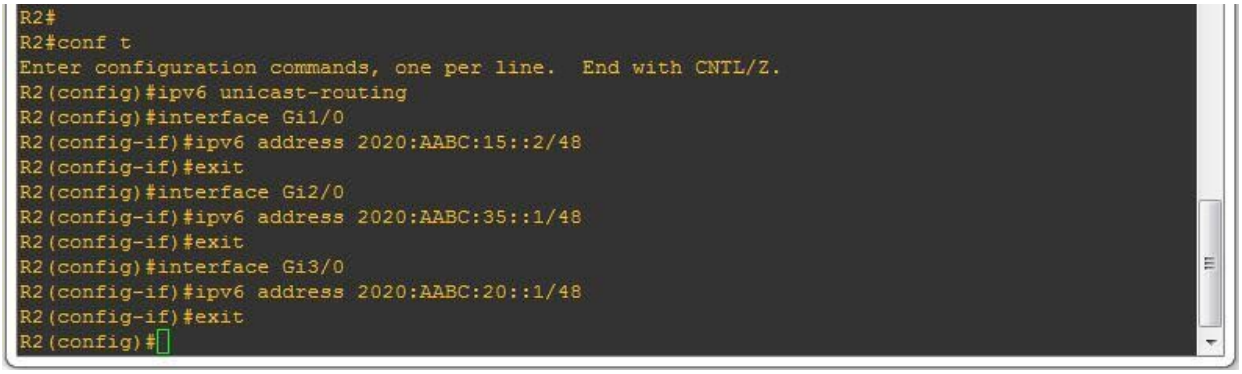


Figura 16.13

Pe R3:

```
R3(config)#ipv6 unicast-routing

R3(config)#interface Fa0/0
R3(config-if)#ipv6 address 2020:AABC:2::1/48
R3(config-if)#no shutdown

R3(config)#interface Gi2/0
R3(config-if)#ipv6 address 2020:AABC:25::2/48
R3(config-if)#no shutdown

R3(config)#interface Gi3/0
R3(config-if)#ipv6 address 2020:AABC:20::2/48
R3(config-if)#no shutdown

R3(config)#interface Gi5/0
R3(config-if)#ipv6 address 2020:AABC:24::1/48
R3(config-if)#no shutdown
```

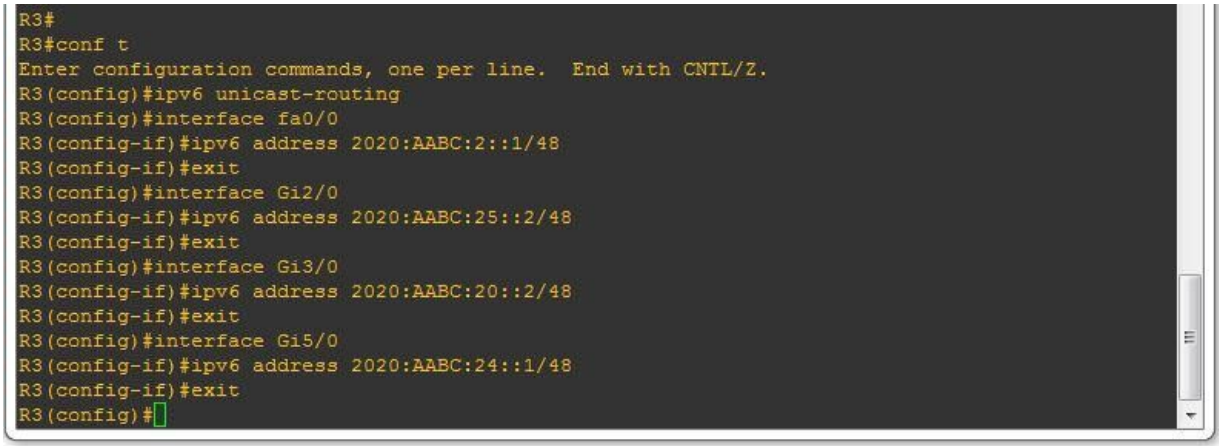


Figura 16.14

Pe R4:

```
R4(config)#ipv6 unicast-routing

R4(config)#interface Fa0/0
R4(config-if)#ipv6 address 2020:AABC:4::1/48
R4(config-if)#no shutdown

R4(config)#interface Gi2/0
R4(config-if)#ipv6 address 2020:AABC:35::2/48
R4(config-if)#no shutdown

R4(config)#interface Gi5/0
R4(config-if)#ipv6 address 2020:AABC:24::2/48
R4(config-if)#no shutdown
```

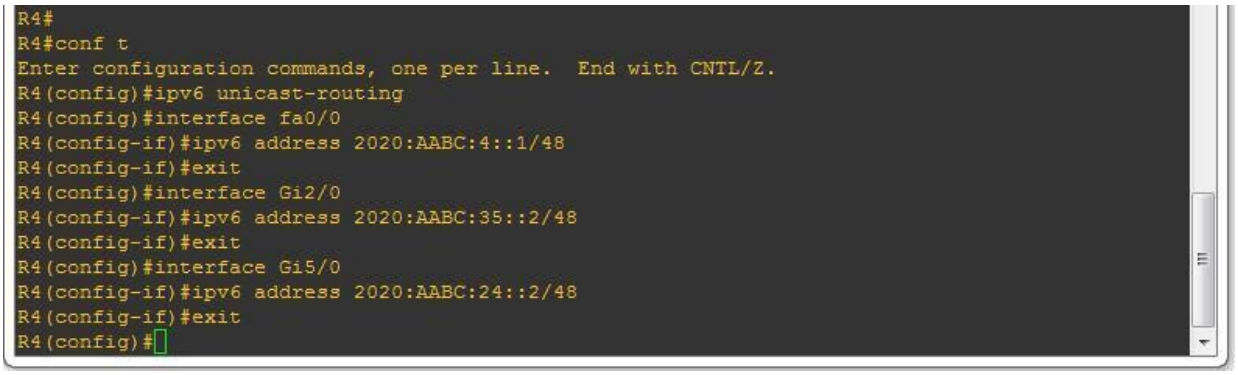


Figura 16.15

Asignarea adreselor IPv6 pe Routere si pe PC-uri a avut loc cu succes, iar acum a sosit momentul sa configuram rutarea intre aceste retele pentru ca dorim sa avem **conectivitate end-to-end**.

Asigurarea conectivitatii end-to-end

Acum ca am terminat asignarea adreselor IPv6, urmeaza **sa asiguram conectivitate end-to-end** (adica din orice punct al retelei, in orice alt punct al retelei). Vom trece prin fiecare element discutat, pana acum: *rute statice si protocoale de rutare (RIP, OSPF, EIGRP)*. Sa reluam topologia:

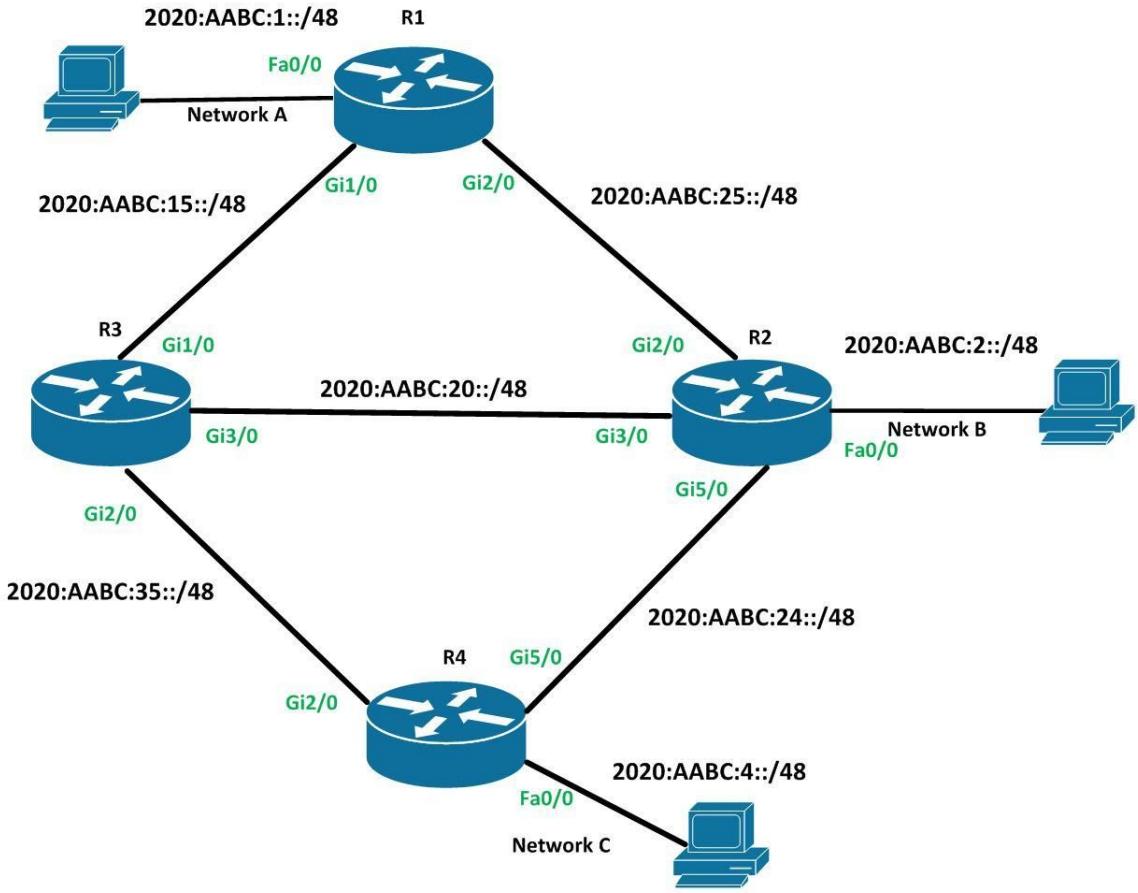


Figura 16.16

1) Configurare Rute Statice pe IPv6

Asadar, ce trebuie sa facem este sa configuram **rute statice**. Iata mai jos, o parte din configul intregii retele. De dragul exemplului, vom seta rutele statice doar pe R1. Pe R1 punem setam rute statice catre fiecare retea:

```
R1(config)#ipv6 route 2020:AABC:24::/48 2020:AABC:15::2
R1(config)#ipv6 route 2020:AABC:20::/48 2020:AABC:15::2
```

```
R1(config)#ipv6 route 2020:AABC:35::/48 2020:AABC:25::2
R1(config)#ipv6 route 2020:AABC:2::/48 2020:AABC:25::2
R1(config)#ipv6 route 2020:AABC:4::/48 2020:AABC:15::2
```

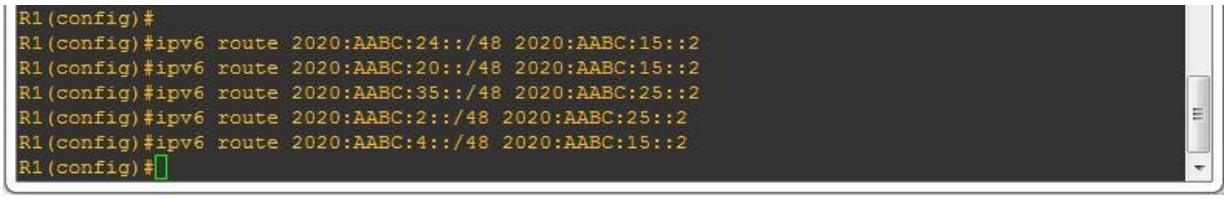


Figura 16.17

lata o parte dintre rutele care au fost create/adaugate in tabela de rutare:

```
R1#show ipv6 route
```

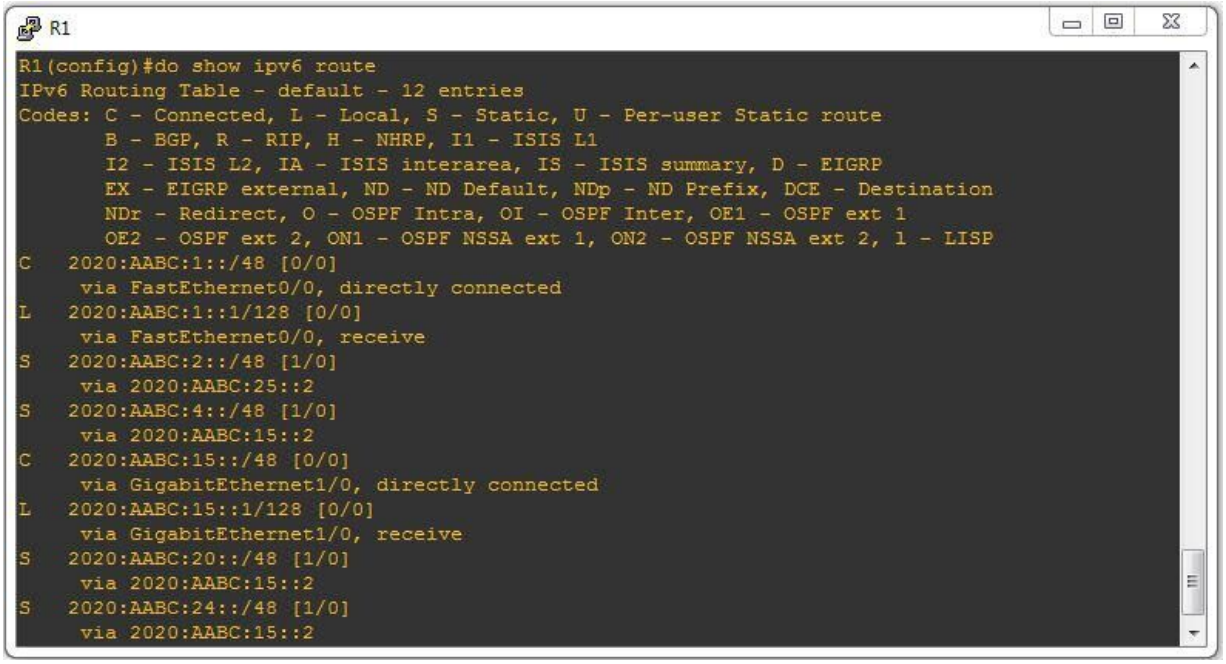


Figura 16.18

2) Configurare RIPng pentru IPv6

Dupa ce am vazut in capitolul 2 ce este RIP si cum se configureaza, este momentul sa-l setam si pe IPv6. In acest caz, el poarta numele de **RIPng** (aka **RIP next generation**) si isi pastreaza toate functionalitatile/capabilitati (rutarea dinamica, update-uri la fiecare 30 de secunde) de la versiunea 2 de pe IPv4. Defapt, RIP pentru IPv6 este *mult* ***mai usor de configurat***.

Iata un exemplu de configurare, pe R1 si R4, din topologia de mai sus (vom omite R2 si R3 pentru ca setarile sunt similare):

Figura 16.19

```
R1(config)#ipv6 router rip NAME

R1(config)#interface Gig0/0
R1(config-if)#ipv6 rip NAME enable
```

Configul va fi similar si pe R4 (vom creea un proces RIPng si il vom porni pe interfete). Te rog sa observi ca denumirea procesului RIPng atat pentru R1 cat si pentru R4 sunt diferite. Nu este necesar ca cele doua sa aiba acelasi nume de proces.

Figura 16.20

Modul in care putem verifica este acelasi (ca in cazul IPv4), singura diferenta fiind modificarea din IP in IPv6:

```
R4#show ipv6 route
```

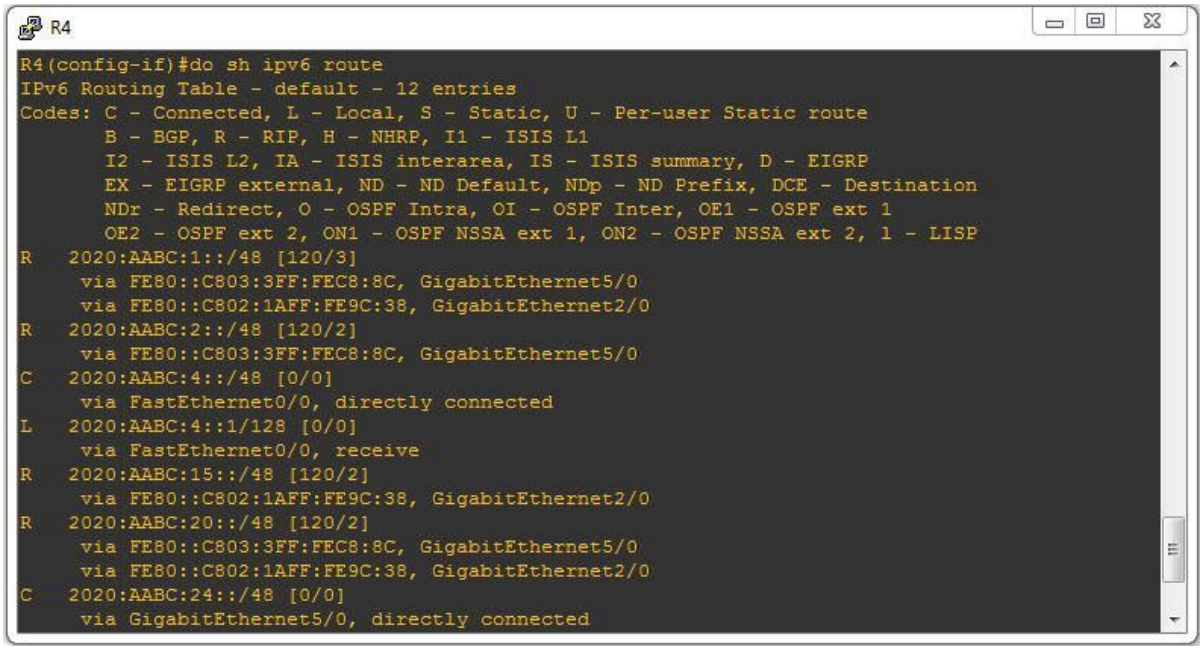


Figura 16.21

`R1#show ipv6 protocols`

Figura 16.22

Spre deosebire de RIP pe IPv4, in figura de mai sus putem vedea interfetele pe care a fost pornit RIPng. Pe IPv4 puteam sa vedem doar retelele (sumarizate) adaugate prin comanda network, iar outputul nu era clar.

3) Configurare OSPF pentru IPv6

La fel ca si in cazul RIP, acum a venit momentul sa activam OSPFv3 (pentru IPv6) pe cele 4 Routere. Inca o data, functionalitatea (OSPF-ului) nu se schimba (se stabilesc relatii de adiacenta intre vecini, Hello-uri la 10 secunde, existenta Ariilor), doar forma de adresare care este pe IPv6.

La fel ca si in cazul RIPng, OSPF este mai usor de configurat. Pentru acest exemplu vom presupune ca toate Routerele se afla in Aria 0. Iata mai jos un exemplu:

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1

R1(config)#interface Fa0/0
R1(config-if)#ipv6 ospf 1 area 0

R1(config-if)#interface Gi1/0
R1(config-if)#ipv6 ospf 1 area 0

R1(config-if)#interface Gig2/0
R1(config-if)#ipv6 ospf 1 area 0
```

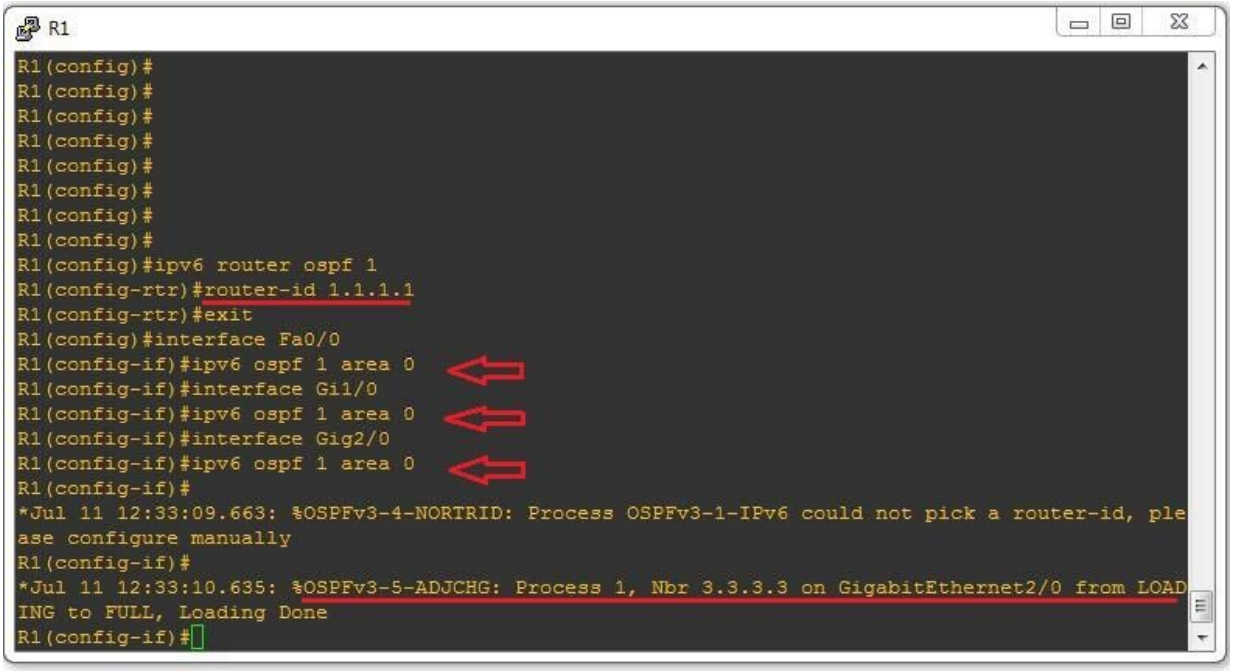


Figura 16.23

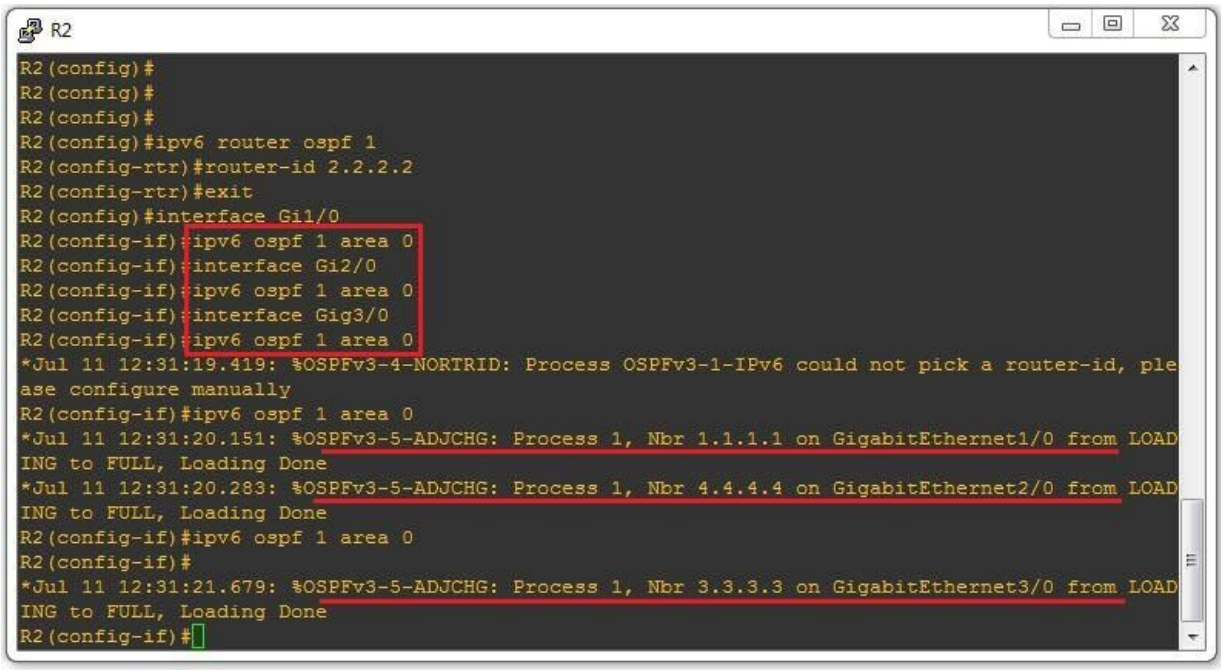


Figura 16.24

lata o parte din rutele invate prin OSPFv3 in tabela de rutare a lui R1:
`R1#show ipv6 route`

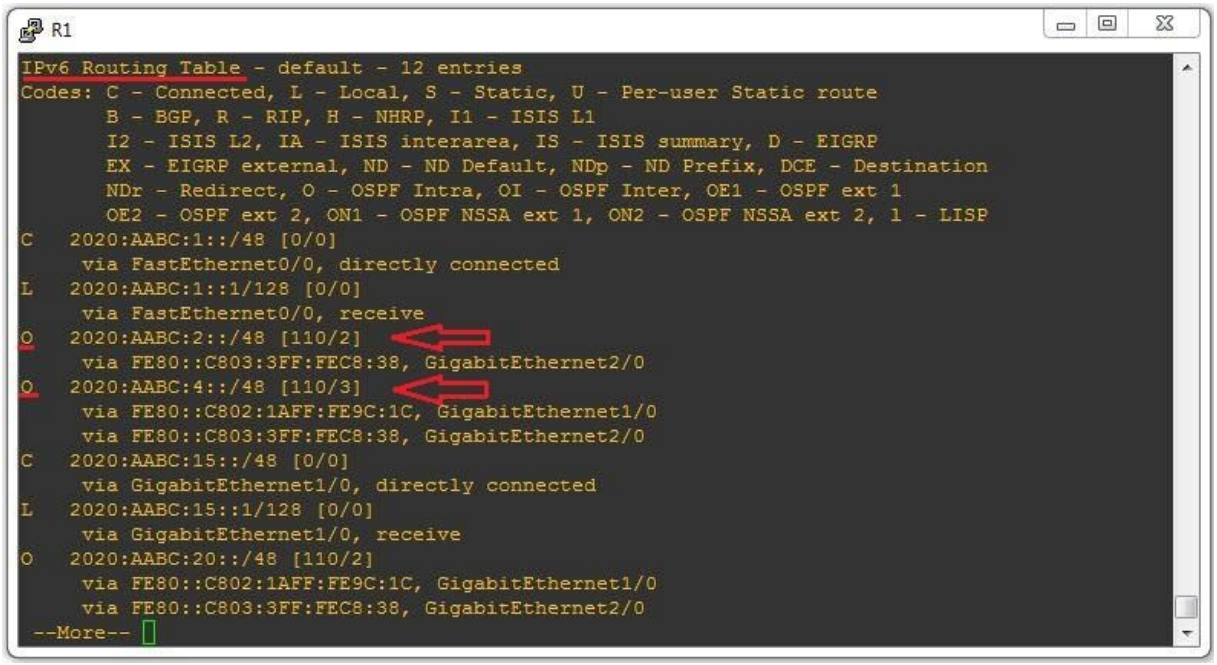


Figura 16.25

`R1#show ipv6 protocols`

In figura de mai jos putem vedea protocolul activ (OSPFv3), interfetele pe care acesta este pornit si aria in care se afla aceste interfete:

```
R1#
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "ospf 1"
  Router ID 1.1.1.1
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet2/0
    GigabitEthernet1/0
    FastEthernet0/0
  Redistribution:
    None
R1#
```

Figura 16.26

`R1#show ipv6 ospf neighbor`

In figura de mai jos putem vedea vecinii lui R1 din topologie (R2 respectiv R3):

```
R1#
R1#show ipv6 ospf neighbor

            OSPFv3 Router with ID (1.1.1.1) (Process ID 1)

Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
3.3.3.3           1   FULL/DR         00:00:30    4               GigabitEthernet2/0
2.2.2.2           1   FULL/BDR        00:00:33    3               GigabitEthernet1/0
R1#
```

Figura 16.27

4) Configurare EIGRP pentru IPv6

Acum am ajuns la cel de-al treilea protocol de rutare studiat de noi, EIGPR. Cand vorbim de Ipv6, acest protocol poarte denumirea de **EIGRPv6** si isi pastreaza toate caracteristicile de la IPv4 (stabilesc relatii de adiacenta intre vecini, Hello-uri la 5 secunde, tabela de topologie, etc.). Acum sa trecem la partea de configurare. Vom configura EIGRPv6 pe cele 4 Routere din topologia initiala:

```
R1(config)#ipv6 router eigrp 123
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#no shutdown
```

NOTE: un aspect important aici, este faptul ca trebuie sa pornim procesul prin `#no shutdown`

```
R1(config)#interface Gig1/0
R1(config-if)#ipv6 eigrp 123

R1(config)#interface Gig2/0
R1(config-if)#ipv6 eigrp 123

R1(config)#interface Fa0/0
R1(config-if)#ipv6 eigrp 123
```

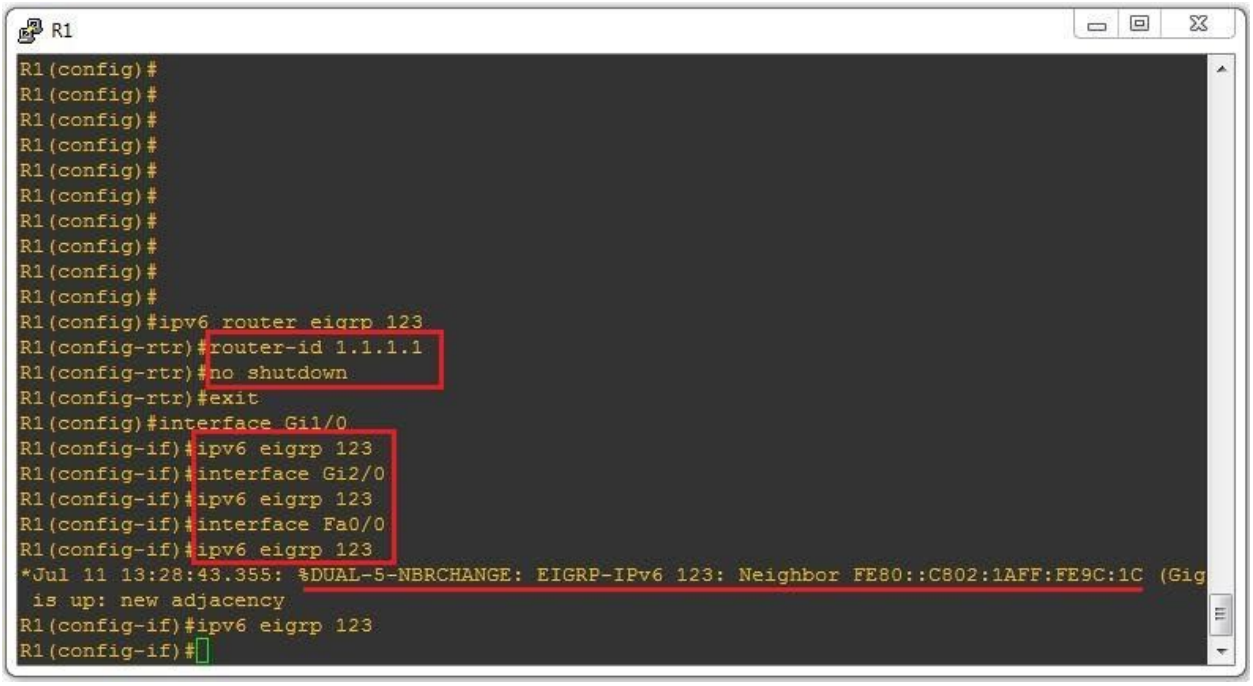


Figura 16.28

Adaugam aceleasi setari pe R3 (si restul Routerelor), si obtinem rezultatul subliniat, exact *adiacenta intre Routere*.

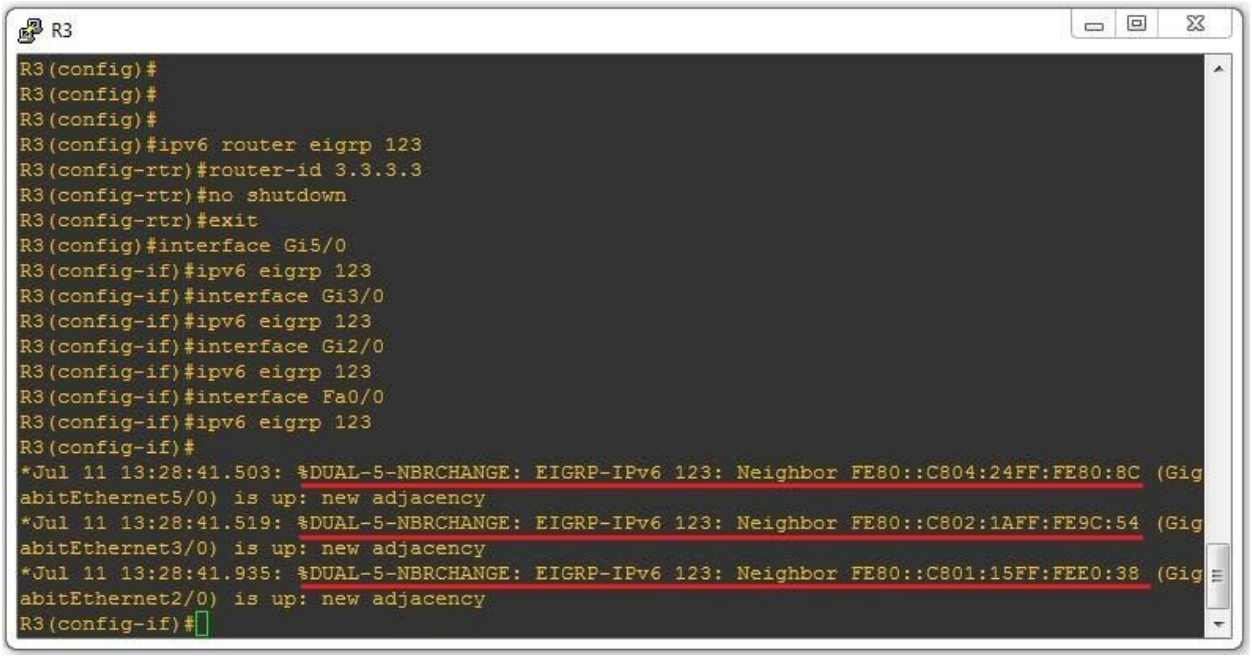


Figura 16.29

`R3#show ipv6 eigrp neighbors`

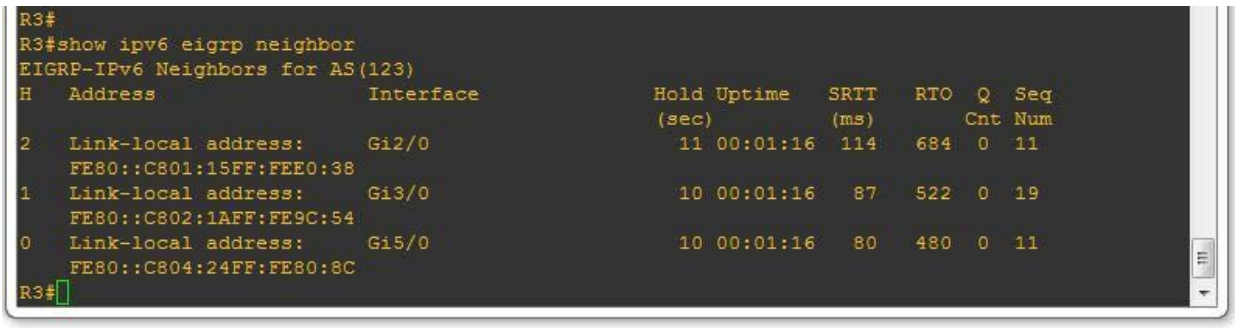


Figura 16.30

In figura de mai sus putem vedea vecinii lui R3 din topologie (R1, R2 si R4), fiecare dintre acestia folosind adresa IPv6 link local pentru comunicarea intre vecini.

`R3#show ipv6 route`

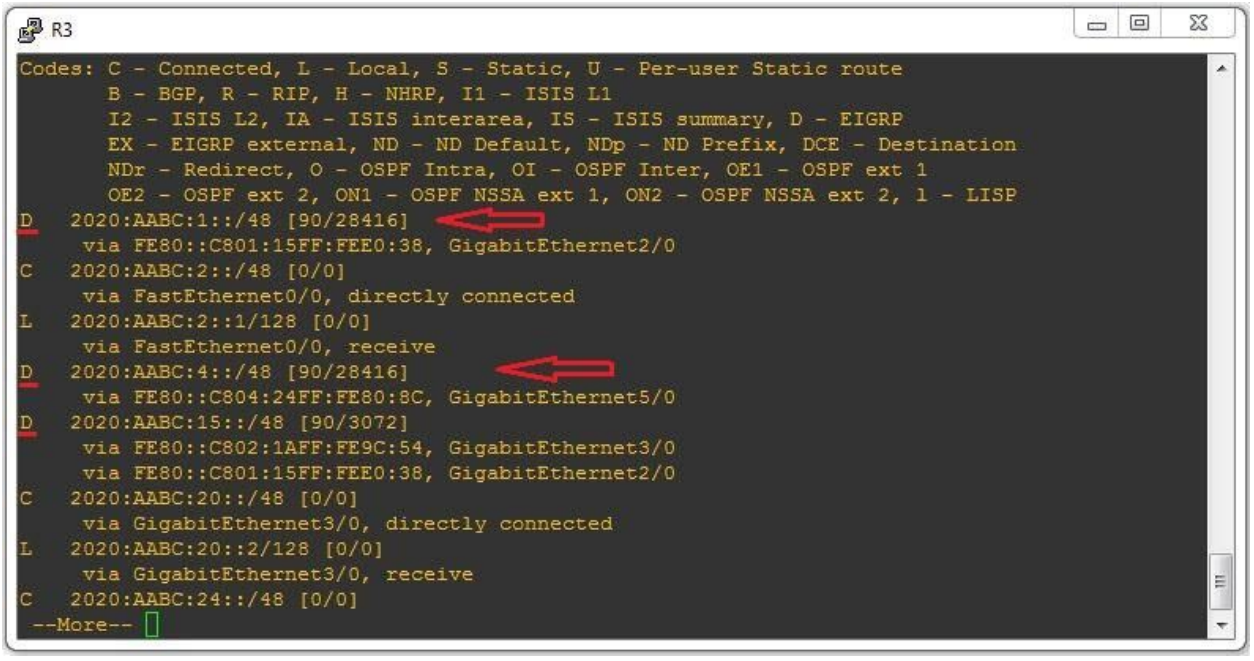


Figura 16.31

```
R3#show ipv6 protocols
```

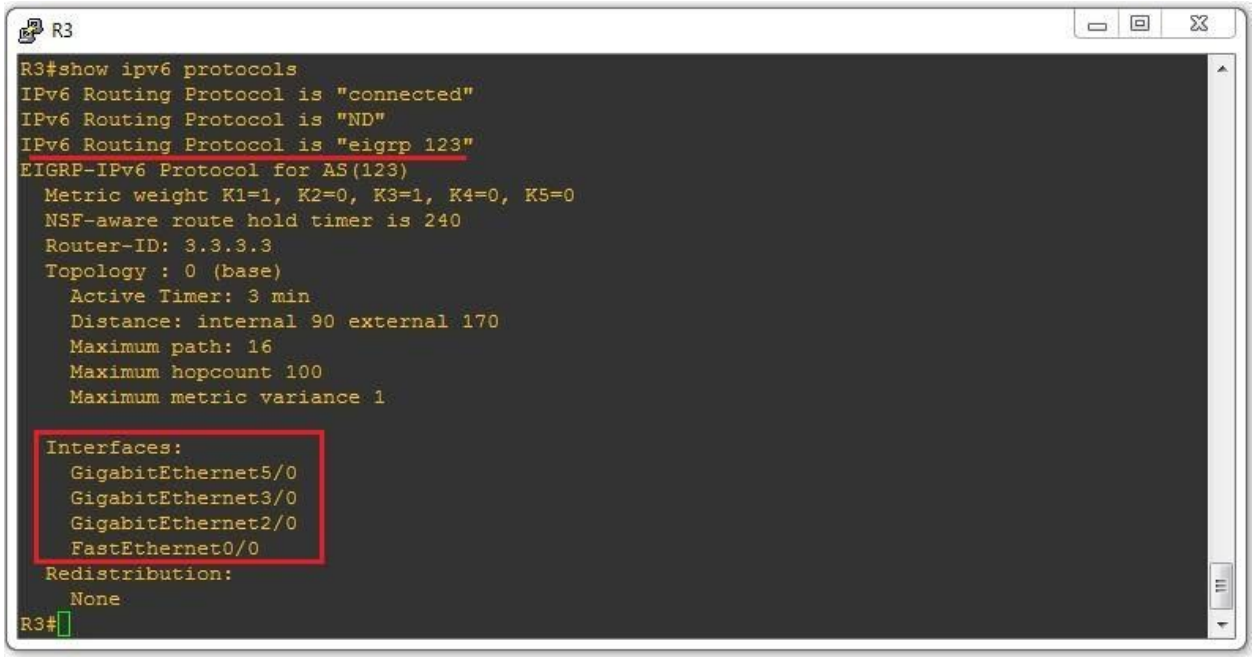


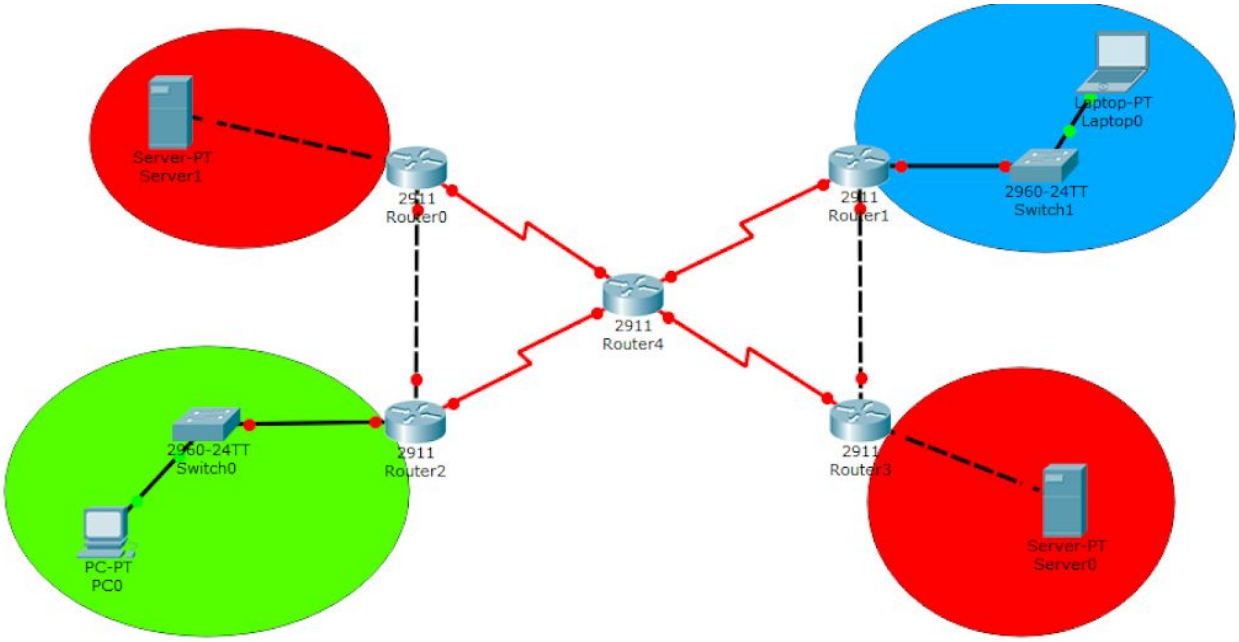
Figura 16.32

In figura de mai sus putem vedea protocolul activ (EIGRP in AS-ul 123), interfetele pe care acesta este pornit si AS-ul in care se afla acestea.

Laboratorul #14

Acum a venit momentul cel mai interesant, configurarea IPv6 pe Routere. Mai intai vom seta adrese IPv6 pe fiecare Router si end-device in parte, dupa care vom configura RIPng si Rutare Statica pentru acestea. Te invit sa urmaresti cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aplicarea conceptelor de rutare statica si RIPng pe IPv6.

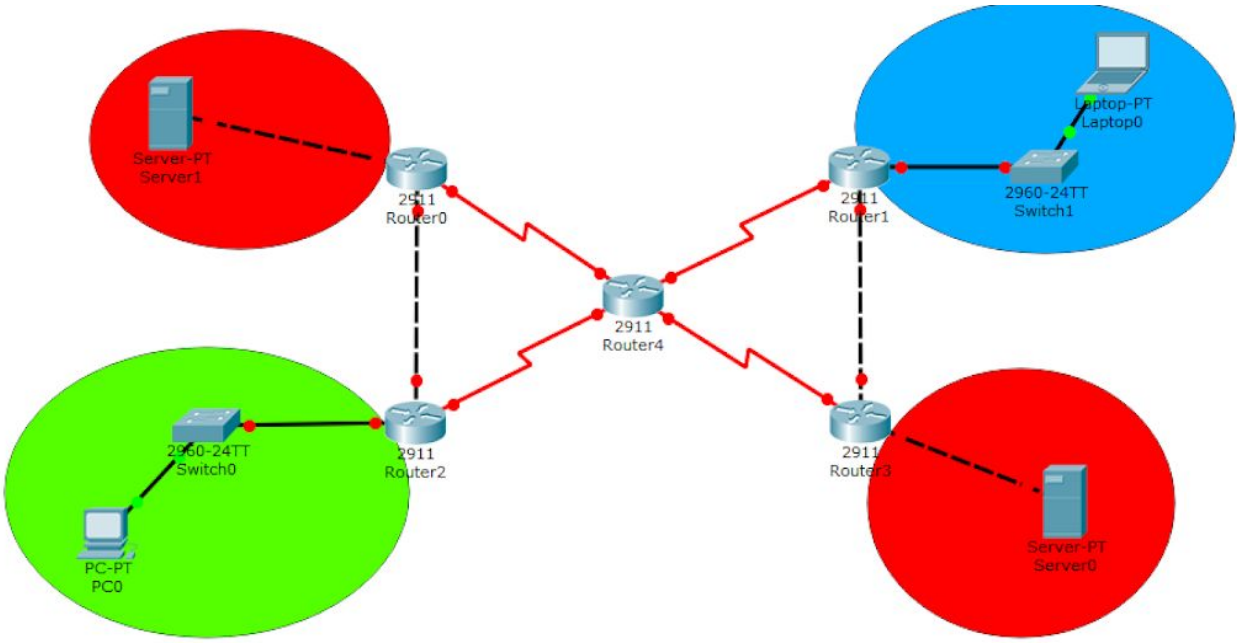


SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !

Laboratorul #15

Vom lua un laborator similar cu cel de mai devreme, dar vom aplica conceptele de OSPFv3 si EIGRP discutate mai devreme. Te invit sa urmaresti cerintele existente in laborator si configureaza corespunzator acestora, dispozitivele existente.

SCOP: Aplicarea conceptelor de OSPFv3 (pe IPv6) si EIGRPv6.



SFAT: Foloseste-te de **manualul de comenzi** pentru a rezolva cu succes exercitiul !